

Aalborg Universitet



## Applied SfM - Capturing and Analysing 3D Surface Data

Nikolov, Ivan Adriyanov

DOI (link to publication from Publisher):  
[10.54337/aau404966619](https://doi.org/10.54337/aau404966619)

Publication date:  
2021

Document Version  
Publisher's PDF, also known as Version of record

[Link to publication from Aalborg University](#)

Citation for published version (APA):  
Nikolov, I. A. (2021). *Applied SfM - Capturing and Analysing 3D Surface Data*. Aalborg Universitetsforlag. <https://doi.org/10.54337/aau404966619>

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

### Take down policy

If you believe that this document breaches copyright please contact us at [vbn@aub.aau.dk](mailto:vbn@aub.aau.dk) providing details, and we will remove access to the work immediately and investigate your claim.



# **APPLIED SfM - CAPTURING AND ANALYSING 3D SURFACE DATA**

**BY  
IVAN NIKOLOV**

**DISSERTATION SUBMITTED 2020**



**AALBORG UNIVERSITY**  
DENMARK



---

---

# Applied SfM - Capturing and Analysing 3D Surface Data

---

---

Ph.D. Dissertation  
Ivan Nikolov

Dissertation submitted September 30, 2020

Dissertation submitted: September, 2020

PhD supervisor: Professor Claus Madsen  
Aalborg University

PhD committee: Associate Professor Georgios A Triantafyllidis (chair.)  
Aalborg University

Associate Professor Jakob Andreas Bærentzen  
Technical University of Denmark

Professor Thomas P. Kersten  
HafenCity Universität Hamburg

PhD Series: Technical Faculty of IT and Design, Aalborg University

Department: Department of Architecture,  
Design and Media Technology

ISSN (online): 2446-1628  
ISBN (online): 978-87-7210-818-6

Published by:  
Aalborg University Press  
Kroghstræde 3  
DK – 9220 Aalborg Ø  
Phone: +45 99407140  
aauf@forlag.aau.dk  
forlag.aau.dk

© Copyright: Ivan Nikolov

Printed in Denmark by Rosendahls, 2020

# Curriculum Vitae

Ivan A. Nikolov



Ivan Nikolov received his Bachelor's degree in Computer Software and Hardware Engineering from in 2012 from the Technical University of Sofia, Bulgaria. He then pursued his Master's degree in Vision, Graphics and Interactive Systems at Aalborg University, Denmark in 2015. He worked as a research assistant before starting his PhD in November 2016 with both the Computer Graphics Laboratory and the Visual Analysis of People Laboratory at the Section of Media Technology, Aalborg University.

As part of his PhD, Ivan worked with Power Curve Aps, where he was a visiting researcher multiple times and Danmarks Nationale Metrologiinstitut (DFM).

His main interests are computer graphics, interactive systems, computer vision and virtual reality, with a focus on 3D reconstruction and mesh analysis. Ivan has been supervising Bachelor's and Master's degree students at Aalborg University, as well as teaching courses.

## Curriculum Vitae

# Abstract

This PhD thesis looks at the problems associated with Structure from Motion 3D reconstruction for object and surface analysis and inspection. The use of 3D data has become an important part of many research fields, because of the additional information and versatility that it provides, compared to more traditional 2D data. Two main parts of the SfM processing pipeline are featured in the thesis, with additional sub-parts in each.

Structure from Motion data capturing plays an important part in the success and quality of the produced 3D reconstructions. Special care needs to be taken in the way images are shot, the capturing setup, used cameras, as well as the processing done to the said images. The thesis contains a number of publications, which present benchmarks and analysis methods for determining the quality and robustness of the produced reconstructions. These benchmarks come with datasets and ground truths, which can be used by others to further the research. In addition, a capturing environment for producing synthetic images is presented, to cut the preparation time and help users that start in the field. The image capturing can be done both terrestrially and in the air, using unmanned aerial vehicles (UAVs), so the thesis presents also a straightforward approach for localization of drones.

Because it relies on 2D images, as input data for creating 3D reconstructions, SfM does not contain a straightforward way to determine the real life scale of the captured objects or surfaces. This is why this thesis presents two ways to determine both the reconstructions' absolute scale and its uncertainty, by using additional sensors and fusing their data with the SfM data. Both position sensors like GPS and distance sensors like LiDAR are used for this and processing pipelines are presented for each.

The produced 3D surface data needs to be also analysed. First, for removing any noise and geometrical errors introduced by SfM and later for determining the quality of the reconstructions and if they can be used for their intended tasks. An automatic method for separating noise and real surface roughness is presented in the thesis, together with a comparison between SfM and microscopy reconstructions, for determining the accuracy of the captured data.

## Abstract

Finally, two use cases for using the reconstruction data for classification of surfaces are presented. The first one classifies the surfaces, depending on the roughness and how close it is to a standardised sandpaper roughness. The second use case is for simulating tactile sensations of different surfaces through changing amplitude and frequencies of a vibro-tactile actuator, using the roughness information captured with SfM.

# Resumé

Ph.d.-afhandlingen undersøger problemstillinger, der er associeret med "Structure from Motion", også kaldet SfM, rekonstruktion af objekter samt overfladeanalyse og inspektion. Brugen af 3D information er blevet en vigtig del af mange forskningsområder på grund af den information og fleksibilitet 3D information tilføjer, sammenlignet med traditionelt 2D data. To primære dele af SfM processen er fremhævet i denne afhandling, med flere underafsnit for hver.

"Structure from Motion" dataindsamling spiller en vigtig rolle for succes og kvaliteten af den producerede 3D rekonstruktion. Der skal udvises særlig omhu i den måde billederne tages på, indsamlingsopsætningen, det brugte udstyr, såvel som behandlingen af de indsamlede billeder. Afhandlingen indeholder et antal publikationer, der repræsenterer sammenlignings- og analysemetoder for at definere kvaliteten og robustheden af de producerede rekonstruktioner. Disse sammenligninger kommer med dataset, og "ground truth" data, der kan blive brugt af fremtidige forskere til deres forskning. Foruden dette, er et indsamlet miljø til at producere syntetiske billeder, der minimerer forberedelsestiden, og som der kan hjælpe nye forskere inden for dette felt, også inkluderet. Indsamlingen af billeder kan både gøres fra jorden og i luften ved brug af ubemandede luftfartøjer (UAV), hvilket er grunden til, at denne afhandling præsenterer en ligetil metode til lokalisering af droner.

SfM har ikke nogen måde at definere den virkelige størrelse på det rekonstruerede objekt eller overflader, da SfM grundlæggende bruger 2D billeder som input data. Dette er grunden til, at denne afhandling præsenterer to måder til at fastslå både rekonstruktionens absolutte skala og dens usikkerhed. Den absolutte skala samt usikkerheden udregnes ved hjælp af ekstra sensorer, hvis data fusioneres med data fra SfM. Både positions sensorer som GPS, samt afstands sensorer som LiDAR, bliver brugt til dette. Begge disse processers metode er præsenteret i denne afhandling.

Den rekonstruerede 3D overflades data skal analyseres, hvilket først gøres ved at fjerne al støj og alle geometriske fejl, som SfM introducerer. Dette er nødvendigt for senere at kunne definere kvaliteten af rekonstruktionen, samt vurdere om den kan bruges til dens dertil definerede brug. En automatisk

## Resumé

metode til at separere støj fra den rigtige overflades ruhed bliver præsenteret i denne afhandling, sammen med en sammenligning af SfM og mikroskopi rekonstruktioner. Sammenligningen af SfM og mikroskopi rekonstruktioner er til for at kunne validere hvor præcis den indsamlede data er.

Afhandlingen afsluttes med to eksempler for brugen af den rekonstruerede data til klassifikation af overflader. Det første eksempel klassificerer overflader i forhold til overfladens ruhed sammenlignet med standardiseret ruhed af sandpapir. Det andet eksempel er simulering af den taktile sensation fra forskellige overflader ved at ændre amplituden og frekvensen af en vibrotaktile aktuator. Denne simulering bruger den information om ruhed, der er indsamlet ved hjælp af SfM.

# Contents

<b>Curriculum Vitae</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>Resumé</b>	<b>vii</b>
<b>Thesis Details</b>	<b>xv</b>
<b>Preface</b>	<b>xix</b>
<b>I Overview of the Work</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Structure from Motion Data Capturing . . . . .	5
1.2 Structure from Motion Data Analysis . . . . .	6
1.3 Thesis Structure . . . . .	7
References . . . . .	7
<b>2 Structure from Motion Data Capturing</b>	<b>11</b>
2.1 SfM Overview . . . . .	11
2.2 Image Acquisition . . . . .	13
2.2.1 Camera Related . . . . .	13
2.2.2 Capturing Environment Related . . . . .	17
2.2.3 Captured Object Related . . . . .	17
2.2.4 State of the Art . . . . .	18
2.2.5 Contributions . . . . .	20
2.3 Data Fusion . . . . .	24
2.3.1 Drone Platform . . . . .	25
2.3.2 Sensors . . . . .	27
2.3.3 State Of the Art . . . . .	28
2.3.4 Contributions . . . . .	29
References . . . . .	32

<b>3</b>	<b>Structure from Motion Data Analysis</b>	<b>39</b>
3.1	Reconstruction Quality Analysis . . . . .	39
3.1.1	Introduction . . . . .	39
3.1.2	State of the Art . . . . .	42
3.1.3	Contributions . . . . .	44
3.2	Reconstructed Surface Classification . . . . .	47
3.2.1	State of the Art . . . . .	47
3.2.2	Contributions . . . . .	48
	References . . . . .	51
<b>4</b>	<b>Additional Project Work</b>	<b>57</b>
4.1	Comparison Tests between SfM and Microscopy . . . . .	57
4.1.1	Testing Wind Turbine Surface Reconstructions . . . . .	57
4.1.2	Testing the Influence of Texture on SfM Reconstructions	59
4.2	Combining SfM Solutions for Improving Reconstruction Quality	60
4.3	LiDAR Intensity Roughness Detection . . . . .	62
	References . . . . .	64
<b>5</b>	<b>Conclusion</b>	<b>65</b>
	References . . . . .	68
<b>II</b>	<b>Structure from Motion Data Capturing</b>	<b>71</b>
<b>A</b>	<b>Benchmarking Close-Range Structure from Motion 3D Reconstruction Software under Varying Capturing Conditions</b>	<b>73</b>
A.1	Introduction . . . . .	75
A.2	Related Work . . . . .	76
A.3	Tested Software . . . . .	77
A.4	Datasets . . . . .	77
A.5	Testing Scenarios and Results . . . . .	79
A.5.1	Main Test Scenario . . . . .	79
A.5.2	Follow Up Test Scenarios . . . . .	84
A.6	Conclusion and Future Work . . . . .	85
	References . . . . .	86
<b>B</b>	<b>Interactive Environment for Testing SfM Image Capture Configurations</b>	<b>89</b>
B.1	Introduction . . . . .	91
B.2	Our Proposed Solution . . . . .	92
B.3	Methodology . . . . .	93
B.3.1	Camera Approximation . . . . .	93
B.3.2	Environment Approximation . . . . .	95
B.3.3	Object Approximation . . . . .	96

B.4	Solution Test and Results . . . . .	97
B.5	Use Cases . . . . .	99
B.6	Conclusion and Future Work . . . . .	100
B.7	Acknowledgements . . . . .	100
	References . . . . .	100
<b>C</b>	<b>LiDAR-based 2D Localization and Mapping System using Ellipse Distance Correction Models for UAV Wind Turbine Blade Inspection</b>	<b>103</b>
C.1	Introduction . . . . .	105
C.2	State of the Art . . . . .	107
C.3	Methodology . . . . .	107
	C.3.1 From Corrected LiDAR Data to Initial Drone Localization	109
	C.3.2 Ellipse Distance Correction Model (EDC model) . . . . .	110
	C.3.3 Calculating Initial Angle Between Drone and Blade . . . . .	112
	C.3.4 LiDAR position filtering . . . . .	113
C.4	System Tests and Results . . . . .	114
	C.4.1 Localization Test . . . . .	114
	C.4.2 Mapping Test . . . . .	116
C.5	Conclusion and Future Work . . . . .	118
	References . . . . .	118
<b>D</b>	<b>How Image Capturing Setups Influence the Quality of SfM Reconstructions for Wind Turbine Blade Inspection</b>	<b>121</b>
D.1	Introduction . . . . .	123
D.2	State of the Art . . . . .	124
D.3	Methodology . . . . .	125
	D.3.1 Structure from Motion Overview . . . . .	125
	D.3.2 Experimental Setup . . . . .	126
D.4	Results . . . . .	130
	D.4.1 2D Shape Analysis . . . . .	131
	D.4.2 3D Surface Roughness Analysis . . . . .	134
D.5	Conclusion and Future Work . . . . .	138
	References . . . . .	140
<b>E</b>	<b>Performance Characterization of Absolute Scale Computation for 3D Structure from Motion Reconstruction</b>	<b>145</b>
E.1	Introduction . . . . .	147
	E.1.1 Object 3D Reconstruction . . . . .	147
	E.1.2 State of the Art . . . . .	148
	E.1.3 Using External Sensor Data for Determining Scale and Noise Sensitivity . . . . .	148
E.2	Methodology . . . . .	149

## Contents

E.2.1	SfM Pipeline . . . . .	149
E.2.2	Least-Squares Transformation Estimation . . . . .	149
E.2.3	Verifying Sensor Readings . . . . .	151
E.2.4	Synthetic Testing Scenarios . . . . .	152
E.2.5	Covariance Propagating of Positioning Noise . . . . .	155
E.3	Real World Testing . . . . .	156
E.4	Conclusion and Future Work . . . . .	159
	References . . . . .	160
<b>F</b>	<b>Calculating Absolute Scale and Scale Uncertainty for SfM using Distance Sensor Measurements</b>	<b>163</b>
F.1	Introduction . . . . .	165
F.2	State of the Art . . . . .	166
F.3	Methodology . . . . .	167
F.3.1	SfM Reconstruction Overview . . . . .	168
F.3.2	Scaled SfM Reconstruction Using Distance Sensors . . . . .	169
F.4	Uncertainty Modeling and Propagation . . . . .	176
F.5	Implementation . . . . .	180
F.6	Testing and Results . . . . .	181
F.6.1	Testing Setup . . . . .	181
F.6.2	Scale Test Results . . . . .	182
F.6.3	Scale Uncertainty Test Results . . . . .	184
F.6.4	Discussion . . . . .	186
F.7	Conclusion . . . . .	187
	References . . . . .	187
<b>III</b>	<b>Structure from Motion Data Analysis</b>	<b>193</b>
<b>G</b>	<b>Rough or Noisy? Metrics for Noise Estimation in SfM Reconstructions</b>	<b>195</b>
G.1	Introduction . . . . .	197
G.2	State of the Art . . . . .	198
G.3	Methodology . . . . .	200
G.3.1	General Mesh-based Metrics . . . . .	201
G.3.2	Capturing Setup-based Metrics . . . . .	206
G.4	Implementation . . . . .	210
G.5	Testing and Results . . . . .	211
G.5.1	Data Gathering . . . . .	212
G.5.2	Correlation Analysis . . . . .	213
G.5.3	Initial Testing . . . . .	215
G.5.4	Subset Testing . . . . .	217
G.5.5	Industrial Context Test . . . . .	220

G.6	Conclusion and Future Work . . . . .	222
	References . . . . .	223
<b>H</b>	<b>High-resolution Structure-from-Motion for Quantitative Measurement of Leading Edge Roughness</b>	<b>229</b>
H.1	Introduction . . . . .	231
H.2	Methods and Materials . . . . .	233
H.2.1	Blade Mock-up . . . . .	233
H.2.2	SfM Capturing Conditions . . . . .	234
H.2.3	SfM Reconstruction . . . . .	236
H.2.4	Replica Moulding . . . . .	238
H.2.5	Confocal microscopy . . . . .	238
H.2.6	Image processing and data analysis . . . . .	239
H.3	Results . . . . .	240
H.4	Discussion . . . . .	244
H.5	Conclusion . . . . .	247
	References . . . . .	248
<b>I</b>	<b>Preliminary Study on the use of Off-the-Shelf VR Controllers for Vibrotactile Differentiation of Levels of Roughness on Meshes</b>	<b>257</b>
I.1	Introduction . . . . .	259
I.2	State of the Art . . . . .	260
I.3	Methodology . . . . .	260
I.4	Experiment and Results . . . . .	263
I.4.1	Experiment Setup . . . . .	263
I.4.2	Participants and Captured Data . . . . .	266
I.4.3	Results and Discussion . . . . .	266
I.5	Conclusion and Future Work . . . . .	267
	References . . . . .	269
<b>J</b>	<b>Quantifying Wind Turbine Blade Surface Roughness Using Sandpaper Grit Sizes</b>	<b>273</b>
J.1	Introduction . . . . .	275
J.2	State of the Art . . . . .	276
J.3	Methodology . . . . .	277
J.3.1	SfM Overview . . . . .	277
J.3.2	Sandpaper 3D Reconstruction . . . . .	278
J.3.3	Sandpaper Feature Extraction . . . . .	279
J.4	Experimental Scenarios and Results . . . . .	283
J.5	Conclusion . . . . .	286
	References . . . . .	287

## Contents

# Thesis Details

**Thesis Title:** Applied SfM - Capturing and Analysing 3D Surface Data  
**Ph.D. Student:** Ivan Nikolov  
**Supervisor:** Associate Professor Claus Brøndgaard Madsen

The main body of the thesis consists of the following papers, divided into groups:

## Structure from Motion Data Capturing

### Image Acquisition

- A Nikolov, I., & Madsen, C. (2016). Benchmarking close-range structure from motion 3D reconstruction software under varying capturing conditions. In Euro-Mediterranean Conference (pp. 15-26). Springer
- B Nikolov, I., & Madsen, C. B. (2019). Interactive Environment for Testing SfM Image Capture Configurations. In VISIGRAPP (1: GRAPP) (pp. 317-322). SCITEPRESS.
- C Nikolov, I., & Madsen, C. (2017). LiDAR-based 2D localization and mapping system using elliptical distance correction models for UAV wind turbine blade inspection. In International Conference on Computer Vision Theory and Applications (Vol. 7, pp. 418-425). SCITEPRESS.
- D Nikolov, I., Kruse, E.K., & Madsen, C. B. (2020). How Capturing Setups Influence the Quality of SfM Reconstructions for Wind Turbine Blade Surface Inspection. - Article accepted by SPIE, awaiting publication

### Data Fusion

- E Nikolov, I., & Madsen, C. B. (2019). Performance Characterization of Absolute Scale Computation for 3D Structure from Motion Reconstruction. In VISIGRAPP (5: VISAPP) (pp. 884-891). SCITEPRESS.

F Nikolov, I., & Madsen, C. B. (2020). Calculating Absolute Scale and Scale Uncertainty for SfM Using Distance Sensor Measurements: A Lightweight and Flexible Approach. In *Recent Advances in 3D Imaging, Modeling, and Reconstruction* (pp. 168-192). IGI Global.

## Structure from Motion Data Analysis

### Reconstruction Quality Analysis

G Nikolov, I., & Madsen, C. B. (2020). Rough or Noisy? Metrics for Noise Estimation in SfM Reconstructions. - Article accepted to MDPI Sensors Journal

H Nielsen, M., Nikolov, I., Kruse, E. K., Garnæs, J. & Madsen, C. B. (2020). High-resolution Structure-from-Motion for quantitative measurement of leading edge roughness. *Energies*, 13(15), 3916

### Reconstructed Surface Classification

I Nikolov, I., Høngaard, J. S., Kraus, M., & Madsen, C. B. (2020). Preliminary Study on the Use of Off-the-Shelf VR Controllers for Vibrotactile Differentiation of Levels of Roughness on Meshes. In *VISIGRAPP (1: GRAPP)* (pp. 334-340). SCITEPRESS.

J Nikolov, I., & Madsen, C. B. (2020). Quantifying Wind Turbine Blade Surfaces Using Sandpaper Grit Sizes - Article in preparation

## Additional Work

As part of the work presented in the thesis a number of datasets have been created:

- Nikolov, I., Madsen, C. (2020), "GGG-BenchmarkSfM: Dataset for Benchmarking Close-range SfM Software Performance under Varying Capturing Conditions", Mendeley Data, v1
- Nikolov, I. (2020), "GGG-BenchmarkSfM- Secondary: Secondary Dataset for Benchmarking Close-range SfM Software Performance Captured with Different Cameras", Mendeley Data, v1
- Nikolov, I., Madsen, C. (2020), "GGG-PositioningSfMScale: Dataset for Testing Scaling and Uncertainty Propagation Through Camera Positioning Data", Mendeley Data, v1

## Thesis Details

- Nikolov, I., Madsen, C. (2020), "GGG - Rough or Noisy? Metrics for Noise Detection in SfM Reconstructions", Mendeley Data, v1

The following papers have been written as part of the PhD work, but are not a part of the thesis, because they explore topics not directly connected to the main body of work:

- Nikolov, I. (2016). Inside-Outside Model Viewing. In Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications: Volume 1: GRAPP (pp. 292-297). SCITEPRESS
- Nikolov, I. (2020). Testing VR headset cameras for capturing written content. In Proceedings of the 23rd International Conference on Academic Mindtrek (pp. 153-156). ACM
- Jensen, T. D., Kasprzak, F., Szekely, H., Nikolov, I., Høngaard, J., & Madsen, C. B. (2020). Preliminary Study on the Influence of Visual Cues, Transitional Environments and Tactile Augmentation on the Perception of Scale in VR - Accepted and awaiting publication
- Henriksen, K., Lynge, M., Jeppesen, M., Allahham, M., Nikolov, I., Haurum, J., Moeslund, T. (2020). Generating Synthetic Point Clouds of Sewer Networks: An Initial Investigation. In Proceedings of the 7th International Conference on Augmented Reality, Virtual Reality and Computer Graphics SalentoAVR (pp. 364-373). Springer

The thesis in all its parts are based on an ongoing research and has been submitted for assessment in partial fulfillment of the PhD degree. The thesis is based on the submitted or published scientific papers which are listed above. Parts of the papers are used directly or indirectly in the extended summary of the thesis. As part of the assessment, co-author statements have been made available to the assessment committee and are also available at the Faculty. The thesis is not in its present form acceptable for open publication but only in limited and closed circulation as copyright may not be ensured.

## Thesis Details

# Preface

The thesis is structured as a collection of papers in partial fulfillment of a PhD study at the Section of Media Technology, Aalborg University, Denmark. The thesis is separated in three parts. The first part consists of introduction to the field, as well as state-of-the-art connected to all the topics covered by the published papers. This first part also contains some of the additional work done as part of the PhD study, which was not published. The next two parts - Structure from Motion Data Collection and Structure from Motion Data Analysis, contain the selected papers published as part of the PhD.

The PhD was part of a larger project "Leading Edge Roughness" carried out from 2015, until 2019. The work done for the thesis was carried out both at Aalborg University, as part of the Visual Analysis of People Lab and the Computer Graphics Group, as well as at one of the participating companies PowerCurve ApS, Aalborg.

I would like to thank my supervisor Claus Madsen, for providing good supervision through my master's degree and PhD study and for helping and encouraging me to continue and evolve my research. He was there through the good and tough parts and always provided solid advice and direction in my studies and work. I would like to give thanks to my colleagues and friends from the Visual Analysis of People Lab and Computer Graphics Group, as well as associate professor Martin Kraus from Aalborg University, Mikkel Schou Nielsen from DFM and Emil Krog Kruse from PowerCurve. With special thank you to Jens Stokholm Høngaard, Anne Juhler Hansen, Kasper Skou Ladefoged, Joakim Bruslund Haurum, Kasper Hald and Brian Bemman, among other for brightening my day and making me laugh when I most needed it.

I am especially grateful for my girlfriend Gergana for her love, support and strength through the years - the light of my life, even in the darkest nights. Finally, a huge thank you to my mother Paolina - without your help and sacrifices, I would not be where I am.

Ivan A. Nikolov  
Aalborg University, September 30, 2020

## Preface

## **Part I**

# **Overview of the Work**



# Chapter 1

## Introduction

Computer vision has rapidly progressed over the past 70 years, starting from a purely experimental and theoretical field in the 1950s, through the first commercial recognition systems in the 1980s, to its revolutionization in the beginning of the 2000s and the 2010s, due to shift in focus on statistical methods and machine learning. Computer vision has the potential to speed up and automate many time consuming or dangerous tasks. As the field grows, so does the need for more diverse data and methods for verifying that data.

Data streams for computer vision can be roughly divided into 2D and 3D, depending on the dimensionality of the captured data. Images, as well as video sequences, disregarding time progression, are examples of 2D data, capturing only vertical and horizontal dimensions from a single perspective. To be able to factor in multiple perspectives of the captured data, a third dimension - most commonly denoted as depth, is necessary.

Capturing 3D information from the real world has always been of great interest for many research fields like medicine [1], geology [2], archaeology [3], the manufacturing [4] and energy industries [5], construction [6] and even the entertainment industry [7]. Unlike capturing 2D information from images, 3D data requires either specialized sensors or additional processing to extract and gather 3D information from 2D streams of data.

This thesis is a part of a larger research project called Leading Edge Roughness, centered around quantifying the amount of erosion on the surface of wind turbine blades. This research is funded as project number EUDP 2015-I under the Danish national EUDP programme. This erosion can then be translated into energy production loss. Knowing this relation between erosion and energy loss, a better understanding when and why blades need to be changed can be achieved. The core of the project required achieving a sub-millimeter surface extraction accuracy and the surface analysis to be performed on-site, while the blades are still in use. This meant that tra-

ditional wind turbine blade damage detection methods using 2D or even depth images, would not be able to extract the necessary roughness information. On the other hand high resolution 3D scanning methods used on decommissioned blades, like capturing microscopy measurements from blade surface molds, were also not feasible because of the required time and effort. A high resolution 3D surface capturing method was required, that could be performed on-site, in sub-optimal conditions. One such method is the photogrammetry solution for capturing the 3D shape of objects - Structure from Motion (SfM). SfM is part of the family of photogrammetry based computer vision algorithms for capturing 3D data and relies on capturing 2D images from multiple positions and using them to reconstruct the 3D scene [8]. Many SfM algorithms presently exist - from commercial [9–11] to open source [12, 13] and the field has had a number of landmark publications [14, 15], that shaped its direction. This thesis was built around tackling the problems that can arise, when using SfM for producing reconstructions for 3D surface inspection.

Each problem is attributed to a number of fields and represents a main chapter of this thesis. A brief summery is given below, followed by the overall structure of the thesis.

A number of hypotheses are given as part of the introduction, which have been an integral part of the PhD project and to which answers are given in the next chapters. These hypotheses are:

1. **It is possible to experimentally map the relationship space between the capturing conditions and setup parameters and the quality of the resultant SfM reconstructions.** This hypothesis is developed based on the need to ensure that the captured images from a wind turbine blade can provide an accurate enough reconstruction of the surface, with minimal geometrical imperfections and noise. The possible conclusions from verifying this hypothesis can be extended for the 3D capturing of every generic object.
2. **Capturing sub-millimeter surface details from wind turbine blade surfaces is possible, provided the used hardware can ensure enough information is gathered and the capturing conditions are met.** Wind turbine blades have shape and surface characteristics that make 3D reconstructing them non-trivial. The information gathered from the work on wind turbine blades, can easily be extended to other reconstruction scenarios in the manufacturing industry, geology, cultural preservation, etc.
3. **It is possible to automatically scale SfM reconstructions of wind turbine blade surfaces to absolute scale, using late data fusion of the reconstructions with positioning or distance data, even if it is not**

## 1.1. Structure from Motion Data Capturing

**possible to manually measure the real surface.** This is an important problem to tackle, as capturing surface details through SfM, cannot normally estimate their absolute real-life scale. This can be especially problematic, when the surface roughness needs to be further analysed. Such analysis can be used for estimating energy loss for wind turbine blades, for capturing the sizes of cracks and defects in manufacturing and building sites, as well as for representing complex and intricate surfaces of historical artifacts.

4. **Surface roughness information and reconstruction noise are separable, using 3D mesh and capturing setup analysis.** SfM can produce high resolution reconstructions, but depending on the capturing conditions and the characteristics of the surface, there can be surface noise and geometric imperfections mixed in with surface details. As the captured surface roughness of the wind turbine blades will be analysed and energy loss calculated from it, it is necessary to limit the amount of captured noise and surface imperfections. The same needs to be true for every use case, where information is extracted from the reconstructed surface or objects. Traditional mesh and point cloud surface analysis can be combined with SfM capturing setup analysis to determine the likelihood that sub-optimal capturing conditions could lead to noise.
5. **It is possible to use the captured 3D surface information for different analysis contexts and for introducing multimodal interactions.** By 3D reconstructing the surface of an object, a digital twin can be created, which can later be used for analysing and quantifying the surface properties. The whole Leading Edge Roughness project is build on the premise that the SfM reconstructions can correctly represent the wind turbine surfaces and can be used to extract aerodynamic properties from them. The verification of hypothesis is equally important for any type of 3D reconstruction use case, where additional information and conclusions, would be extracted from the resultant object or surface. This data can also be used to convey additional information about the reconstructed surface using multimodal actuators.

## 1.1 Structure from Motion Data Capturing

This part of the thesis deals with the possible problems and optimizations in capturing image data for SfM, ensuring optimal 3D reconstruction results. The research here focuses on two main areas:

- Image acquisition in real life - how varying capturing conditions can influence the 3D results and how to test these conditions using a simulated environment

- Data fusion between images and other means of capturing distance data for achieving absolute scale of the reconstructions

Both of these areas are inspired by non-trivial problems that can arise in the SfM processing pipeline. Because SfM is so reliant on the quality of the captured images, the output 3D reconstructions can dramatically vary depending on the amount of images, overlap between captured images, image views from which the captured 3D surface is seen, amount and positioning of lighting and the use of additional equipment like turntables. In addition, the large number of possible SfM solutions makes it hard to choose the most appropriate program, depending on the use case and the balance between robustness versus imperfect capturing setups and quality of the resultant reconstructions. To properly evaluate these, a number of testing scenarios and metrics are devised. These are tested both for real life data acquisition, as well as using a simulated environment, together with approximated real life lighting and camera capture setups.

The reliance on only 2D image data also introduces other problems, such as the inability to achieve automatic absolute real life scale of the reconstructed objects and surface. Although some trivial solutions are present for remedying this, like manual measurements and the use of markers, a robust and automatic way of scaling is needed. A number of distance measurement sensors are tested - both GPS, laser based and ultra sound based and their output is used for determining the real life scale of the reconstructions. The uncertainty of the sensors is also taken into consideration and the ways it influences the calculated scale uncertainty is explored.

## 1.2 Structure from Motion Data Analysis

The second part of the thesis focuses on analysing the created 3D reconstructions. The reconstructed surface shape and structure need to be evaluated, to see if they can successfully capture the real life surfaces. To do this, initially the output of SfM models is compared to traditional means of surface details capturing - metrological microscopy. This research aims to verify that the captured micro 3D roughness structures by SfM are representative of the real life surface and to see how much of them are lost, compared to a very precise, but laborious process like microscopy analysis.

Once the data accuracy can be verified, the next steps are to try to find methods for automatic segmentation of proper surface roughness data and capturing and reconstruction noise. This is again a non-trivial problem, as the severity and position of noise of a reconstructed surface, can vary widely depending on the capturing setup, the real life object's surface properties, the illumination of the scene, etc. For that purpose, a number of custom features are tested for noise detection by classical statistical methods.

### 1.3. Thesis Structure

The final step of the reconstruction pipeline is the actual use cases of the captured 3D data. Because of the very applied nature of the project, not all work on the use cases was scientifically relevant and thus not featured in the thesis. Two research use cases are developed, which will be discussed. The first data use case is for classification of the surface roughness, as standardized sandpaper roughness values. This way surface micro structures can easily be compared and verified. The second use case is for using the captured surface roughness, to reproduce the tactile sensation of the real life objects, through modulated vibrations. This way additional modalities can be used when presenting 3D reconstructed information.

## 1.3 Thesis Structure

The research featured in the thesis is formed as a direct verification of the posed hypotheses, as well as a reaction to some of the problems and requirements faced during the research. In addition to these, some of the applied work, which has not resulted in publications has been described in Chapter 4. The thesis is divided into an introductory part and papers part.

The introductory part gives an overview to the field of SfM, as well as the problems related to it separated in the 4 research areas - Image Acquisition, Data Fusion, Reconstruction Evaluation and Surface Information Extraction. The introductory part also contains the state of the art for each of the 4 research areas. As part of the research necessary for developing the thesis, a number of datasets were created. The introductory part will also contain some information on them, as it was deemed necessary to present the datasets, as many of them are in the field of wind turbine blade inspection and overall object digitalization using SfM, where full datasets containing ground truth are always needed. Each research area section ends with a list of conclusions and observations. A general conclusion section is added at the end of the introductory part, which aims to summarize the work, presented in all research areas.

After the introduction, the next two parts contain all the papers, written as part of the PhD study. Four additional papers were written, during the employment time, which are not directly connected to the overarching topic of the thesis and as such are not included in the thesis.

## References

- [1] L. M. Galantucci, M. G. Guerra, and F. Lavecchia, "Photogrammetry applied to small and micro scaled objects: A review," in *International Conference on the Industry 4.0 model for Advanced Manufacturing*. Springer, 2018, pp. 57–77.

## References

- [2] A. M. Cunliffe, R. E. Brazier, and K. Anderson, "Ultra-fine grain landscape-scale quantification of dryland vegetation structure with drone-acquired structure-from-motion photogrammetry," *Remote Sensing of Environment*, vol. 183, pp. 129–143, 2016.
- [3] J. B. López, G. A. Jiménez, M. S. Romero, E. A. García, S. F. Martín, A. L. Medina, and J. E. Guerrero, "3d modelling in archaeology: The application of structure from motion methods to the study of the megalithic necropolis of panoria (granada, spain)," *Journal of Archaeological Science: Reports*, vol. 10, pp. 495–506, 2016.
- [4] Q. Péntek, S. Hein, A. Miernik, and A. Reiterer, "Image-based 3d surface approximation of the bladder using structure-from-motion for enhanced cystoscopy based on phantom data," *Biomedical Engineering/Biomedizinische Technik*, vol. 63, no. 4, pp. 461–466, 2018.
- [5] D. Zhang, R. Watson, G. Dobie, C. MacLeod, A. Khan, and G. Pierce, "Quantifying impacts on remote photogrammetric inspection using unmanned aerial vehicles," *Engineering Structures*, p. 109940, 2020.
- [6] S. Tuttas, A. Braun, A. Borrmann, and U. Stilla, "Acquisition and consecutive registration of photogrammetric point clouds for construction progress monitoring using a 4d bim," *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 85, no. 1, pp. 3–15, 2017.
- [7] N. Statham, "Use of photogrammetry in video games: a historical overview," *Games and Culture*, p. 1555412018786415, 2018.
- [8] T. Luhmann, S. Robson, S. Kyle, and J. Boehm, *Close-range photogrammetry and 3D imaging*. Walter de Gruyter, 2013.
- [9] Agisoft, "Metashape (photoscan)," <http://www.agisoft.com/>, 2010, accessed: 2020-06-15.
- [10] Bentley, "Contextcapture," <https://www.bentley.com/en/products/brands/contextcapture>, 2016, accessed: 2020-06-15.
- [11] 3dFlow, "3df zephyr," <http://www.3dflow.net/>, 2011, accessed: 2020-06-15.
- [12] C. Sweeney, T. Hollerer, and M. Turk, "Theia: A fast and scalable structure-from-motion library," in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015, pp. 693–696.
- [13] E. Rupnik, M. Daakir, and M. P. Deseilligny, "Micmac—a free, open-source solution for photogrammetry," *Open Geospatial Data, Software and Standards*, vol. 2, no. 1, pp. 1–9, 2017.

## References

- [14] C. W. VisualSFM, "A visual structure from motion system," 2011.
- [15] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

## References

## Chapter 2

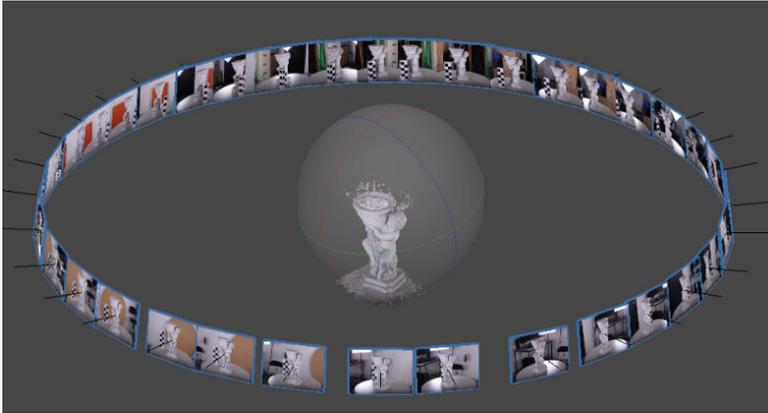
# Structure from Motion Data Capturing

This chapter starts with a brief overview of the SfM pipeline. After that, it presents the problems connected to data capturing for Structure from Motion and the state-of-the-art for both testing and evaluation. In addition, this chapter looks at ways for data fusion for solving the problem of scaling the reconstructed objects to real life absolute scale. As the solving of these problems is essential for achieving high quality reconstruction results, this chapter is invaluable for understanding the concepts of SfM.

### 2.1 SfM Overview

SfM is a part of the family of photogrammetry based reconstruction methods. In its core SfM is a multi-view photogrammetry method, which relies on capturing of image data from different positions of an object or surface and using that data to capture the whole 3D shape. As shown from the research of [1, 2], images need to be taken from both different positions and orientations, with a high degree of overlap, in both horizontal and vertical direction. This is done because surface features need to be seen from different positions and later on matched between images. This process can be seen in Figure 2.1.

The captured features can come from traditional computer vision feature extractors and descriptors like SIFT [3] or SURF [4]. The camera settings information like focal length and resolution, as well as the EXIF camera positioning and rotation information are then used to compute the intrinsic and extrinsic matrices for each image taken. These are then combined with the matched image features to simultaneously compute the a sparse point cloud of the imaged object or surface, together with precise camera positions. For



**Fig. 2.1:** Example of captured image positions in for an SfM reconstruction

doing this, techniques like bundle adjustment can be used [5]. This is an iterative method, which is highly dependent on the quality and quantity of the provided input data. The resultant sparse points are transformed into a dense point cloud by projecting the camera views onto the object and interpolating the information. The details captured on the dense point cloud are thus explicitly tied to the details the cameras could capture, when taking the images of the object. A number of post-processing steps can then be used to mesh the point cloud, to smooth out noise from it or to calculate a high resolution color texture using the image information. The different output steps in the SfM processing pipeline can be seen in Figure 2.2.

## 2.2. Image Acquisition

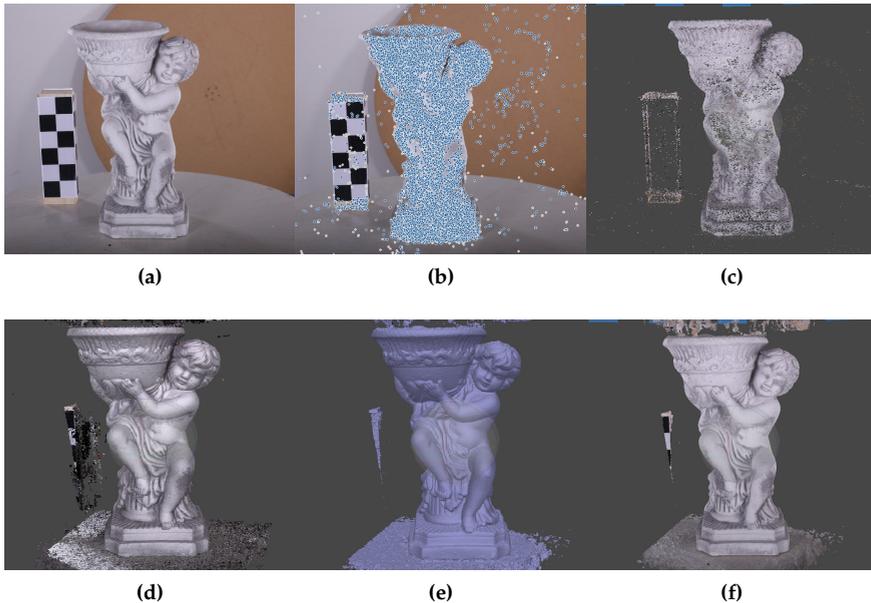


Fig. 2.2: Different output steps of the SfM pipeline, starting with an input image 2.2a, finding and matching features 2.2b, creating the sparse point cloud 2.2c, interpolation and creation of the dense point cloud 2.2d, meshing of the point cloud 2.2e and computing the texture from the input images 2.2f

## 2.2 Image Acquisition

As mentioned in the previous section, the results from SfM are highly correlated to the quality of the image acquisition setup. This setup can be separated into three main categories:

- Camera related - depending on the camera quality and settings, as well as camera positions and orientation.
- Capturing environment related - the illumination of the environment, the use of a static versus a dynamic capturing setup, the use of a background, etc.
- Captured object related - dependent on the "difficulty" of capturing of the selected object, such as the object's surface, color, shape and size.

### 2.2.1 Camera Related

The choice of camera can be extremely important for the quality of the produced SfM reconstruction. The most straightforward measurement of the

chosen camera is its resolution, because it directly limits how much information can be extracted for the reconstruction. Because most SfM methods are directed to work offline, time is not regarded as an issue, when processing high resolution images. This only leaves the computer's processing power as the main bottleneck.

### Camera resolution

All feature extractors and descriptors are limited by the fidelity, with which the captured surface is presented in the images. This fidelity can be dependent on many factors - the camera resolution, the distance between the camera and the imaged object or surface, the focal length of the used camera lens. A measurement of these limitations is the ground sampling distance (GSD). This is the measurement for how much is the real life distance equivalent of the distance between two pixel centers on the captured image [6].

$$GSD = \frac{S_w H}{F_r W_{im}} \quad (2.1)$$

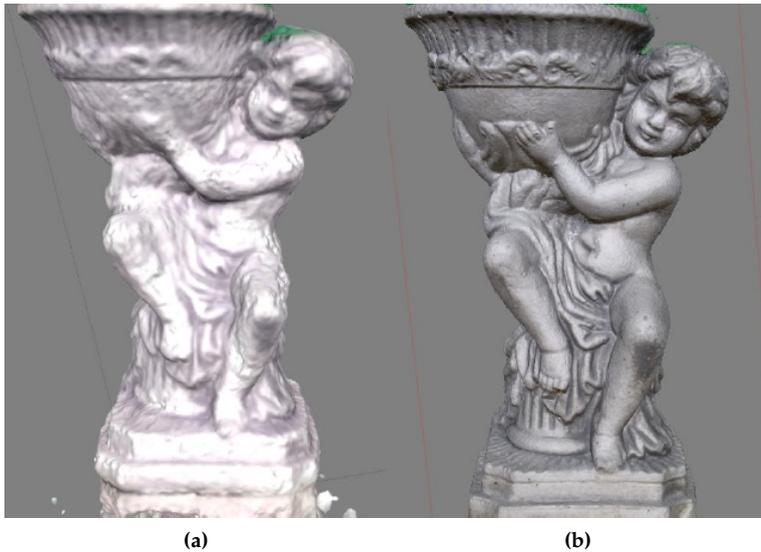
To calculate the GSD, Equation 2.1 can be used where  $S_w$  is the sensor width of the camera in  $mm$ ,  $F_r$  is the focal length of the camera in  $mm$ ,  $H$  is the distance between the camera and the photographed surface and  $W_{im}$  is the image resolution width. To get a better understanding of the difference that the resolution of the photo can make two examples are calculated - one with images taken from a camera with a resolution of 2208x1242 and one from a camera with a resolution of 5472x3648. Two of the other variables are set the same - 2 meters distance and 70 mm focal length lens, with the only other difference being the difference in sensor size. The resultant 3D reconstructions are shown in Figure 2.3, where it can clearly be seen how much detail is lost in the first example.

### Camera sensor

The camera sensor size is closely related to the camera resolution and can also have adverse effects on both the capturing setup and the reconstruction output. The camera sensor captures all the light coming from the camera lens, when the shutter is open, meaning that the quality of the captured environmental information of the image is dependent on it. The size of the camera sensor can vary from a full frame 35mm large sensor, to various cropped sensors. As the sensor grows a number of factors change, that can affect the SfM image acquisition:

- The larger the sensor size, the larger the field of view of the camera is, meaning that less images are needed to capture the whole object, with a high degree of overlap.

## 2.2. Image Acquisition



**Fig. 2.3:** Difference between SfM reconstructions using images with different resolutions - 2.3a uses images with resolution of 2208x1242, while 2.3b is reconstructed from images with resolution of 5472x3648

- The larger the sensor size, the more light and information it can gather, meaning that less quantization noise is present in the produced image, which can affect negatively the reconstruction. The larger sensor size can also lead to larger size of its cells, meaning that more light can be gathered, leading to better light sensitivity. This sensitivity is translated to more details in overly dark or bright spots, which can be represented in the reconstruction.
- The larger the sensor size, the narrower the depth of field, meaning that parts outside of focus become blurrier. This can both help with removing details from unwanted parts of the images, but can also mean that more images need to be taken for the best results while reconstructing.

### Camera settings

The camera settings can also have an effect on the 3D reconstruction. Each of the three main settings in the exposure triangle [7] - aperture, shutter speed, ISO, can have an effect on how much light the camera can process, leading to under or over exposure of the resultant image. The exposure can also be viewed as an additive system, as seen in the APEX exposure approximation [8], where the exposure can be represented using Equation

2.2, where  $A_v$  is the aperture,  $T_v$  is the shutter speed,  $S_v$  is the ISO number and  $B_v$  is the environmental brightness.

$$E_v = A_v + T_v = B_v + S_v \quad (2.2)$$

On top of that each of the settings can have additional effects on the image depending on the value used. Low shutter speed can introduce blurring to the captured images, removing possible features, used by SfM. A high ISO can cause a high amount of noise in the images, which can produce false features and end up degrading the final 3D reconstruction. Wider aperture settings can cause a shallow depth of field, meaning that less of the image is in focus, while narrower aperture has the opposite effect. Another thing to take into account is that if the aperture becomes too small, the effects of light diffraction become apparent, which can cause blurring. Diffraction can cause some of the light rays coming into the camera, to be bent and having to travel more than the ones that are not diffracted and this can cause interference on the image. This is most noticeable in very small details, as they become blurred because of the interference between the diffracted and non-diffracted light rays.

### Camera lens

If more detail needs to be capture for the SfM reconstruction, but the distance to the object or surface cannot be changed, another way to achieve a higher fidelity is by changing the user camera lens. The higher the focal length of the lens the farther away images can be taken of a surface or object, capturing the same amount of detail. On the other hand the higher the focal length the smaller the field of view (FoV) of the lens, as well as the depth of field (DoF). Both of these can be calculated, when the focal length of the lens is known. Field of view can be calculated using Equation 2.3 , where the angle of view (AoV) is calculated using Equation 2.4 and  $D_s$  is the distance to the captured object,  $S$  is the camera sensor width and  $f$  is the focal length.

$$FoV = 2 \tan\left(\frac{AoV}{2}\right)D_s \quad (2.3)$$

$$AoV = 2 \arctan\left(\frac{S}{2f}\right) \quad (2.4)$$

While the depth of field can be calculated using Equation 2.5, for the distance to the far and near focus planes  $FP$  and Equation 2.6 for calculating the hyperfocal distance  $D_H$ , which is the distance between the camera lens and the closest in-focus object, when the lens is focused at infinity. Here  $A_v$  represents the aperture, while  $CoC$  is the circle of confusion constant.

## 2.2. Image Acquisition

$$FP = \frac{D_H D_s}{D_H \pm (D_s - f)} \quad (2.5)$$

$$D_H = \frac{f^2}{A_v CoC} \quad (2.6)$$

### 2.2.2 Capturing Environment Related

The capturing environment can also have strong influence on the quality of the produced SfM reconstruction. One of the most important distinctions for the SfM capturing process is if it is done in an indoor or outdoor environment. As part of the thesis work, both indoor and outdoor image capturing is done and a number of pros and cons are seen for both cases. The indoor environment can be very structured, meaning that each part of the setup, illumination and positioning can be carefully monitored and can remain stationary and without change. A possible negative for indoor capturing is the necessity for more lighting, which needs to be also very even, as each part of the surface or object needs to receive relatively the same amount of light. On the other hand outdoor environments are subject to more changes, that can drastically alter the capturing setup, like varying lighting conditions, moving background objects, etc. But outdoor environments can use the sun as a light source, which helps with the even direct illumination.

Another distinction in the capturing environment is if the image object or surface is stationary and the camera is moved to achieve different views or when the camera is stationary and the object or surface are moved. The first case is very useful for large surfaces or objects and is generally more robust to different background environments. The second case makes it easy to capture smaller objects in more confined spaces, but changes in the background environment can ruin the SfM reconstruction. The capturing setup also directly ties together with the necessity to use a screen as a background, when possible, as this eliminates background problems when using a moving object setup and makes masking easier if necessary as a pre-processing step, before the SfM reconstruction.

### 2.2.3 Captured Object Related

The object that is being captured can also have adverse effects on the SfM pipeline. Objects or surfaces can have physical characteristics that makes it extremely hard or even impossible to properly reconstruct them using SfM. In such cases additional techniques are needed, such as capturing epipolar plane images [9] or using noise pattern projection [10]. Some of the most problematic object surfaces for SfM are:

- Transparent surface - as transparent objects refract and reflect light different depending on the viewing positions, capturing and matching features between images becomes almost impossible.
- Reflective surfaces - the same problem as transparent objects, but additional specular highlights can remove detail from the surface and hide possible surface features.
- Monochrome and featureless surfaces - surfaces which are textureless and do not contain pronounced features make finding enough features for matching per image a problem.
- Repeated pattern surfaces - surfaces which contain a texture, which does not change through the whole object, make it easy for the feature descriptor to produce a large number of false positive matches.
- Thin or flat surfaces - surfaces which exhibit rapid changes of how much is seen from them, depending on the viewing position, require a large number of images for proper reconstruction.

#### 2.2.4 State of the Art

Photogrammetry can be summarized as a way to capture and read physical information from the environment, through the use of images or any type of recordings. It has been a relevant research topic in one way or another far before even traditional analog photography. It started as a tool to develop topological maps, capturing landmark buildings and express perspective of 3D items in flat drawings [2, 11, 12]. With the emergence of analog photography and an easier way to capture single and multiple images from different directions of objects, its use expanded into creation of digitized terrains and surface, and the acquisition of precise visuals. Later on in the 1980s close range photogrammetry, started to emerge as a means for capturing data for surface inspection of bridges, tunnels, dams and building facades [13, 14].

Modern day photogrammetry can be categorised in many ways. Depending on the size of the structures captured it can go from satellite and aerial photogrammetry capturing vast parts of the land, to terrestrial photogrammetry for measuring of specific location and down to close-range, microscopic and medical photogrammetry focused on progressively smaller objects. Another way to separate it is by the amount of captured images - going from older single-image photogrammetry, to stereo-photogrammetry capable of describing more physical and depth characteristics of the environment and up to multi-view or multi-image photogrammetry, used for capturing the whole 3D structure of objects and surfaces [2, 15].

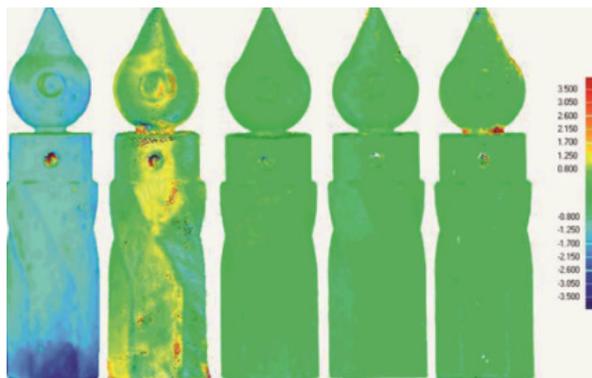
In the state-of-the-art photogrammetry has become an integral part of many industry and research workflows like medicine [16], forensics [17],

## 2.2. Image Acquisition

forestry [18], manufacturing [19], entertainment [20]. Multi-view photogrammetry and especially the bundle adjustment and triangulation variety, that is Structure from Motion have experienced an incredible surge in the last years.

SfM in particular has mostly transitioned from a purely research computer vision topic into viable products, which come as out-of-the-box solutions, ready made for use by non-professionals. Many of the parts of the SfM pipeline can be seen in the landmark research papers by [21–23], responsible for the necessary parts for image position triangulation and optimization and sparse point cloud computation. While algorithms like the ones proposed in [24], lay the groundwork for dense point cloud reconstruction and detailed visual representation.

These workflows were later on translated to many state-of-the-art SfM solutions. Some of these are black box, commercial solution, which do not offer an easy way to change the underlying computer vision algorithms, to suit different use cases. Such software solutions are Metashape by Agisoft [25], ContextCapture by Bentley [26], RealityCapture by Capturing Reality [27], 3DF Zephyr by 3Dflow [28], etc. On the other hand there are many open source SfM solutions, which give different levels of granular control of the underlying algorithms to the end users like MicMac [29], COLMAP [30] and Meshroom [31].



**Fig. 2.4:** Example of testing object reconstructions using different SfM methods and 3D sensors [32]

Even though the development of SfM algorithms is seen as a solved scientific problem, the research around capturing, pre-processing and analysing the results from it is still going strong. Important problems still exist, like the optimization of the capturing setups for producing highly accurate 3D reconstructions, benchmarking the performance of algorithms on different objects and surface (Figure 2.4), as well as developing testing environments. Different benchmarks were created depending on the areas, where SfM is used, like specific ones for archaeology [33], geology [34] or for inspection

purposes in the wind turbine industry [35]. Other benchmarks relied on more general datasets, to look at the overall performance of commercial and open source application and to find the strengths and weaknesses of each. Examples of these can be seen in the work of [36] and [32], where image datasets are used to test the performance of SfM algorithms in close-range capturing scenarios. Other research like the paper proposed by [37], extends this idea by providing custom self-captured datasets, as well as an online website, where people can upload the results of their own algorithms in a makeshift leader board. Capturing images for SfM can become a daunting task, especially having to consider all the possible pitfalls like camera position, orientation, settings, illumination and environmental setup. This is why researchers like [38] and [39], have developed pipelines for creating synthetic images for testing of SfM solutions (Figure 2.5).

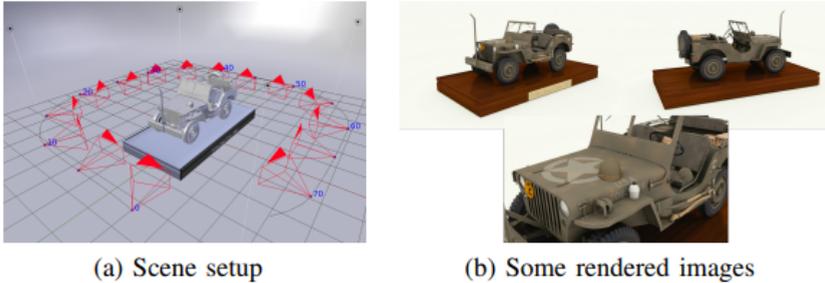
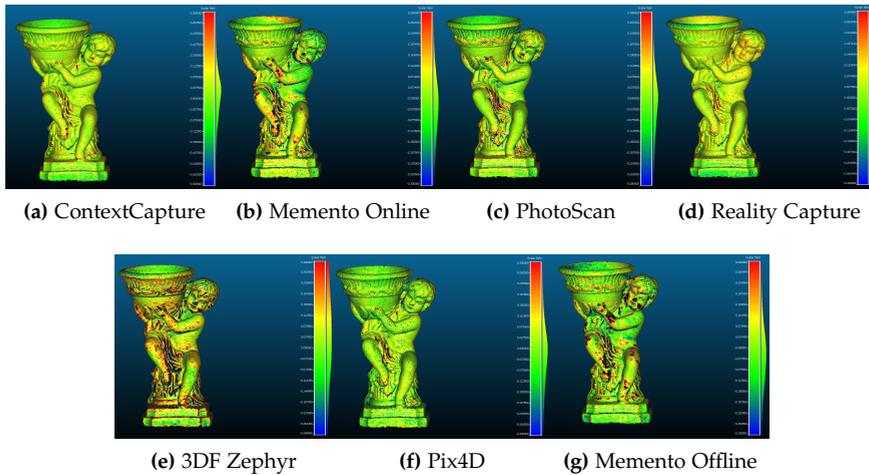


Fig. 2.5: Example of a Blender environment for capturing synthetic images for SfM reconstruction [38]

### 2.2.5 Contributions

The research presented in this section was done as ground work for the thesis research and as a reaction to the lack of robust and extensive overviews, benchmarks and workflows for testing SfM solutions. The overwhelming amount of SfM applications and pipelines and the fact the vast majority of them are presented as expensive blackbox software solution, makes it hard to judge them on equal ground and to decide which use cases are best suited to which software.

## 2.2. Image Acquisition



**Fig. 2.6:** Reconstructed objects using different SfM solutions and close they are to a ground truth object [40], paper A

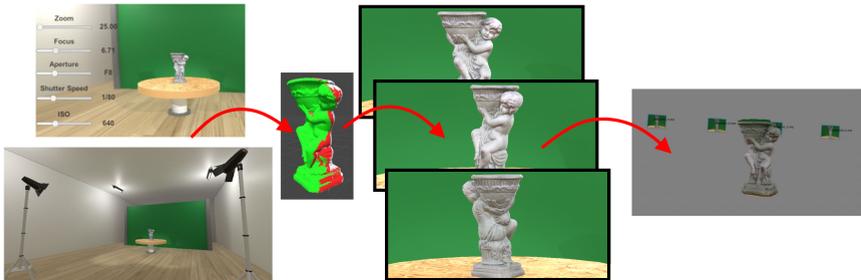
An important publication as part of the thesis was "*Benchmarking close-range structure from motion 3D reconstruction software under varying capturing conditions*" [40] (presented as paper A). It presents a comprehensive benchmark on all the state of the art commercial SfM software solutions 2.6. An important part of this study is the fact that it focuses on many aspects of the capturing setup - the captured object, the positions and orientations of the images, the use of a turntable and the lighting setup. This a useful resource for comparison of SfM solutions, as the number of general and varied benchmarks focusing on both variations in capturing conditions and object surfaces is limited. A number of conclusions were made based on the publication:

- Using images with large overlap and varying horizontal and vertical angular coverage results in better reconstructions.
- Using a stationary object and a moving camera, produces better reconstructions than moving objects on a turntable.
- Correct lighting conditions play an important part in producing noise free reconstructions.
- Commercial SfM software solutions can fall in two categories - ones fairly robust to sub-optimal capturing conditions, but producing lower detail reconstructions and ones producing high-detail surfaces, but prone to failure when presented with worse capturing conditions.

As part of this research two comprehensive image datasets were created, together with ground truth 3D models made with a high accuracy white

light scanner from *Aicon*. The first dataset [41] was comprised of the data described in the paper. The second dataset [42] was made from all the data that could not go in the paper and contains images taken with different types of cameras, as well outdoor and indoor image sets.

The laborious process of creating the datasets, showed that there could be interest in developing an easy to use application for setting up SfM capturing environment and taking synthetic images. This way the heuristic process of choosing the best possible setup for a particular object can be streamlined. Such a software was developed in Unity and presented in "*Interactive Environment for Testing SfM Image Capture Configurations*" [43] (presented as paper B). The thing that separated this paper from the state of the art applications, is the fact that it presented a real-time capturing environment, that users can potentially move around it and change to suit their needs. The captured images were also created with an approximated illumination model, so rendering times were almost non-existent. The conclusions from the paper were that even with the approximated illumination model the produced results can be useful for setting up initial capturing and it could cut the testing time significantly. The process is shown in Figure 2.7.



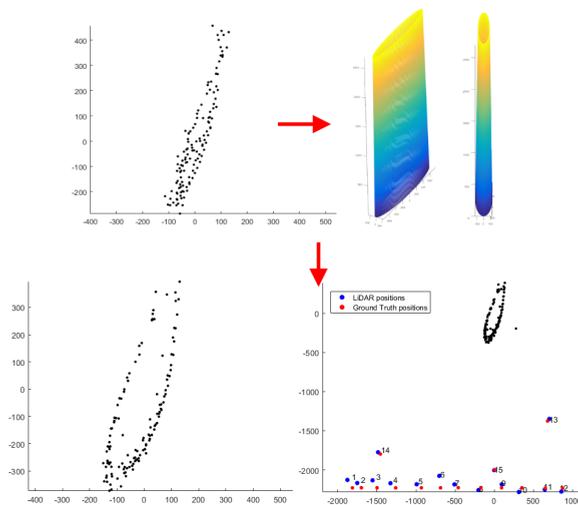
**Fig. 2.7:** The Unity interactive environment for capturing synthetic images for SfM presented in [43], paper B

As the main project, that the PhD was connected to, was detecting leading edge roughness on wind turbine blades, a drone system needed to be created for capturing images of on-site, in-use wind turbine blades. These capturing conditions called for the development of semi-autonomous self-positioning and localisation drone algorithms. This was a non trivial task, because of the glossy, featureless surface of wind turbine blades, which prevents self-positioning using traditional camera based Simultaneous Localization and Mapping (SLAM) algorithms. This led to a research into using LiDAR data for drone flight self positioning and achieving the necessary image capturing patterns. Because of the limited processing power on the drone, a lightweight LiDAR algorithm using prior information about the wind turbine blade shape was developed. This resulted in the publication of the pa-

## 2.2. Image Acquisition

per "LiDAR-based 2D Localization and Mapping System using Ellipse Distance Correction Models for UAV Wind Turbine Blade Inspection" [44] (presented as paper C). This paper was presented as an easier and more straightforward alternative (Figure 2.8) to traditional simultaneous localization and mapping (SLAM) [45, 46] and flight path planning [47, 48] approaches. The research path, showed two main conclusions:

- Prior information about the simplified shape of the blade can be utilized, as part of a localization algorithm.
- Additional sensors can provide valuable data that can be used together with images, as a multimodal input to SfM.



**Fig. 2.8:** The positioning and localization algorithms using ellipse prior information, as shown in [44], paper C

This led to the final paper, presented in this section - *How Capturing Setups Influence the Quality of SfM Reconstructions for Wind Turbine Blade Surface Inspection* [49], which is awaiting publication from SPIE and presented as paper D in the thesis. The paper takes what was learned in the previous localization and mapping research and combines it with the testing paradigms presented in paper A. The scenarios focus on combinations of different image overlaps, positions and distances from a wind turbine blade and compare both the representation of the overall blade shape, as well as the roughness of its surface. The paper demonstrates how the reconstructed surface of the blade degrades with the changes in the capturing setup (Figure 2.9). Three conclusions follow from this paper:



The second reason is that because of its input of purely 2D images, SfM cannot easily determine the absolute scale of the reconstructed structures. Additional information needs to be given to the algorithm pipeline for that. This can be done, in the form of proper camera calibration data containing both an intrinsic and extrinsic data for each image taken. Another possibility is to manually measure parts of the real life surfaces and objects and scale the reconstructed representations of the same parts, until the measurements coincide. A third possibility is to use additional sensors to capture real life data and fuse that with the SfM reconstruction, using it to calculate the absolute scale of the object or surface.

### 2.3.1 Drone Platform

Even though the work on the drone platforms did not result in scientific publications, understanding it is necessary for bridging the gap between the capturing setups research and the sensor fusion research. Sensors like the GPS and LiDAR used for developing the drone platforms, were later used for providing additional data to the SfM reconstructions for scaling and uncertainty verification. This is also why these explanations are not set as part of the additional work chapter 4, as the drone platform development is viewed as a more integral part of the research process of the thesis.



Fig. 2.10: Initial testing drone - the Iris+, together with its sensor payload

Through the course of the project, a number of drone platforms were created. They were tested in both indoor and outdoor environments. As part of the outdoor testing the systems were used both on replicas and on real wind turbine blades. Three main systems were developed, as part of the main project and a number of positioning sensors were tested on each. Two of the platforms needed to be build from scratch, while the third one was a more out of the box commercial platform. The first testing system, used to verify the stability and performance of the used positioning and distance

sensors was a small Iris+ drone, shown in Figure 2.10. This drone could not carry the camera equipment required to capture images, but could support all the necessary sensors. The second UAV was a DJI S1000 large payload drone, which could use the same systems, but could also carry the necessary camera equipment. The drone can be seen in Figure 2.11. The two systems were mainly used on mock-up styrofoam wind turbine blades to both test the sensors, but also perform initial 3D reconstructions.



**Fig. 2.11:** The second testing drone - DJI S1000, used for initial experiments containing both the capturing camera and the GPS and distance sensors

For the real wind turbine blade testing a third system was selected, as the previous two were not stable enough. The off the shelf commercial solution by DJI - the Matrice 600 was chosen (Figure 2.12).



**Fig. 2.12:** The final UAV platform - the DJI Matrice 600, which was used for the real life image capturing and sensor test and as part of [50]

The UAV could be used with more precise sensors and finer granularity of control, for both safely flying around the wind turbine blades, but also maintaining the necessary positions for proper 3D reconstruction. The UAV

was tested both on a real wind turbine blade pieces simulating a real scenario, as well as on site of a wind turbine farm.

### 2.3.2 Sensors

A number of sensors were tested for both capturing the distance between the camera and the reconstructed surfaces, as well as positioning and rotation sensors. For positioning a number of GPS sensors were tested, as they could provide a 3D position of the drone and the camera in a real world coordinate frame. This type of sensor can be useful for both the positioning of the drone, as well as giving initial extrinsic camera positions for the captured images used for SfM. The GPS works by leveraging positioning information from satellites orbiting the Earth and triangulating the position of the tracker based on that. Two GPS solutions were tested - a Pixhawk 2 PIX GPS and the DJI Matrice GPS. Both sensors gave consistent readings, but had low precision, making the calculated position move and drift with as much as 1 meter at a time. On top of that the height calculations became worse the closer to the ground the sensors were. One way to improve significantly both the accuracy and precision of the GPS readings is to use a real time kinematic positioning on top of the GPS. This sensor helps to stabilize the GPS readings, by adding a point of reference on the ground close to the GPS tracker. One such sensor is the GPS DRTK-2 from DJI, used as part of the project.

Another type of sensor researched as part of the study was the inertial measurement unit (IMU) sensor, for capturing the precise real world orientation of the capturing platform. The sensor works by employing a combination of an accelerometer, gyroscope and a magnetometer, together with a barometer. The combination between these sensors can give the correct real world X,Y,Z orientation. Initial research was made on IMU sensors, by using the BNO-055 sensor and later on by using the built-in sensors in the drone platforms.

Finally, a number of distance measurement sensors were research, as a means to correctly capture the space between the capturing platform and the imaged surface. A number of sensors, seen in Figure 2.13 were tested - from cheap ultrasound distance sensors, to high resolution 2D LiDAR sensors. The ultrasound distance sensor works by letting out a number of sound pings, listening for their echo and calculating the distance by the amount of time it took them to return. The LiDAR solutions tested in the project, use a similar method, by first sending a number of laser lights in a single or multiple directions and then capturing and measuring the reflected light. These sensors could be useful for providing additional information, which can be later on fused with the SfM image information, to provide a better result and calculate the scale of the reconstruction.

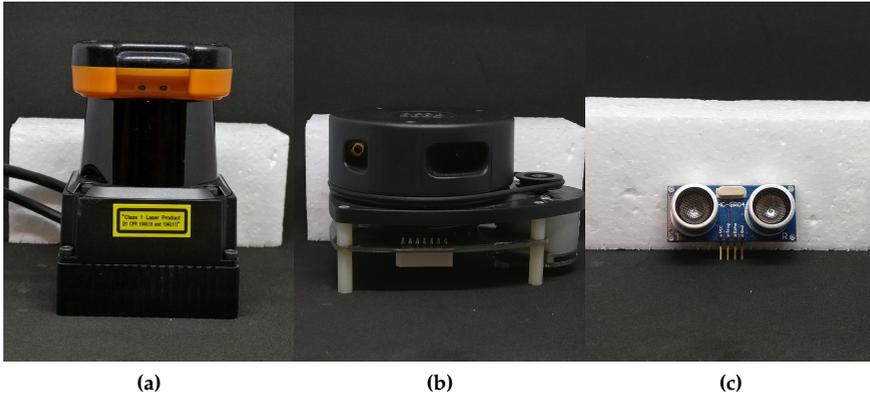


Fig. 2.13: Different distance measurements sensors [51] – LiDAR Hokuyo UTM-30LX (Figure 2.13c), LiDAR rplidar-A1 (Figure 2.13b) and ultrasonic distance sensor HC-SR04 (Figure 2.13a)

### 2.3.3 State Of the Art

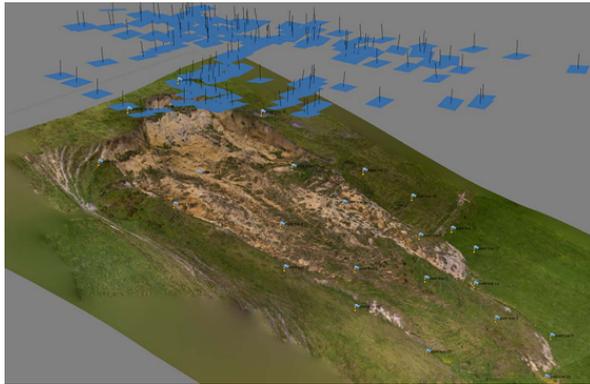


Fig. 2.14: The use of control points as markers for SfM reconstruction and capturing real life scale [52]

As mentioned previously, a lot of the state of the art SfM applications contain manual ways to measure parts of the reconstructed object or surface and then compare those measurements with the ones captured from the real world one and calculate the resultant scale. Another method, that is widely used especially in geological and forestry SfM research [52, 53] is the use of markers with known sizes. This method has the added benefit to also produce high quality reconstructions, but requires manual placement of large enough markers on the surfaces that will be reconstructed and then detecting the markers (Figure 2.14). This approach can be also very time consuming

and out right impossible in some cases, where there is no easy access to the imaged surfaces.

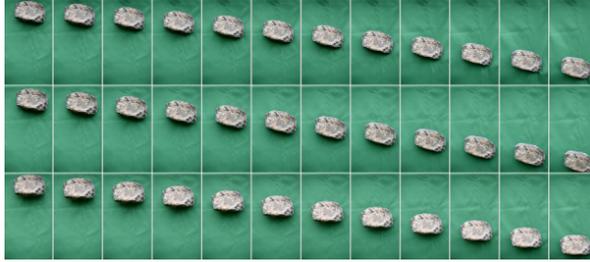


Fig. 2.15: The use of free fall reconstructing the shape and size of a object [54]

Other proposed solutions for calculating the absolute scale of reconstructions use prior information about the camera movements like the model of the motion [54] (Figure 2.15) or the changes in the height [55, 56]. The changes in the position [50, 57] and orientation [58, 59] of the capturing camera can also be used.

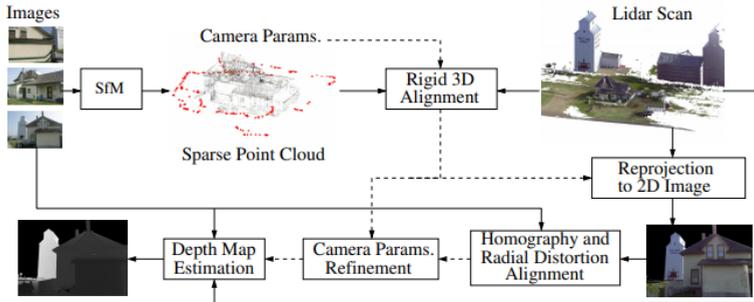


Fig. 2.16: Combining the output from a full LiDAR scanning and SfM data [60]

Another possibility for capturing the absolute scale is by creating a full 3D reconstruction using both SfM and another type of 3D scanning solution, like LiDAR scanning [60] (Figure 2.16), stereo cameras [61] or time-of-flight cameras [62]. In a post-processing step the highly detailed SfM reconstruction, can be matched to the lower detailed, but precisely scaled and positioned secondary scans using means like an iterative closest point algorithm (ICP) [63].

### 2.3.4 Contributions

During the study, a large number of tests were conducted using different sensors and drone platforms. An initial testing scenario was conducted as part of the paper "*Performance Characterization of Absolute Scale Computation for*

"3D Structure from Motion Reconstruction" [50] (presented as paper E) for using positioning information supplied by a GPS to calculate the absolute scale of a 3D reconstruction. Positioning data was captured from both only a GPS and a GPS and RTK fusion and a least-squares transformation estimation [64] was employed to calculate the transformation matrix between the image positions calculated from the SfM algorithm and the real life ones (Figure 2.17).

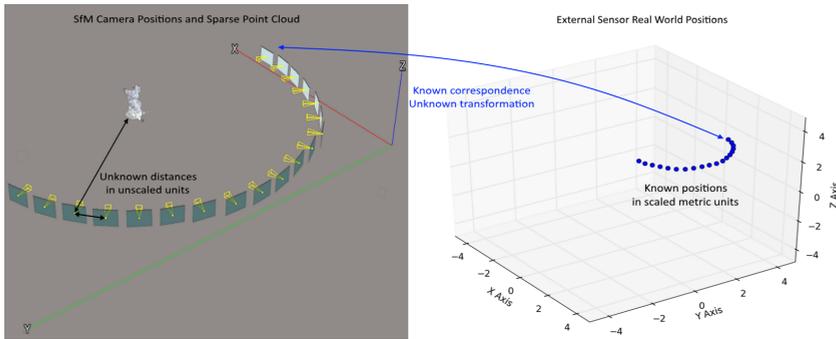


Fig. 2.17: Going from the SfM calculated positions with unknown scale to the captured real life GPS positions [50], paper E

Our study showed, that the results were better than the state of the art algorithms used as part of the commercial SfM applications. Another aspect that was tested as part of the study was the data uncertainty that the sensor could introduce and how that uncertainty translates to scaling errors. To test this, a method is proposed, leveraging the covariance propagation method [65], previously used to detecting uncertainty in pose estimation [66]. It was demonstrated that the sensor uncertainty that propagated through the calculated scale, can be quantified. The dataset used for making this paper, was also made available online for easier verification and reproduction of the results [67]. Two conclusions are taken from the paper:

- Scale uncertainty becomes more problematic, the closer the scale of the measured surface features is to it. When measuring sub-millimeter surface roughness the relatively lower quality of positioning sensor data, like the one from a GPS can be detrimental to the achieved scaling results.
- Positioning sensors cannot account for the captured object's movement or position change, which can be especially problematic in something like a wind turbine blade, which can move dramatically between images.

These conclusions showed that more research was needed, which was published in the book chapter "*Calculating Absolute Scale and Scale Uncer-*

### 2.3. Data Fusion

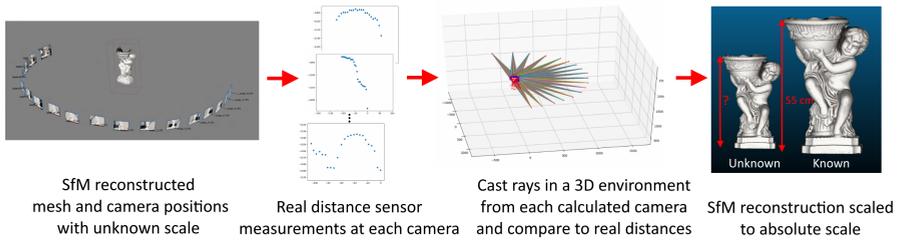


**Fig. 2.18:** The LiDAR sensor connected to the DSLR camera used for capturing the testing data [50], paper E

*tainty for SfM using Distance Sensor Measurements*" [51] (presented as paper F). This research builds upon the findings of paper E [50] and tests out the feasibility of using distance sensors for providing additional real world measurements. These measurements have the advantage of being in the same coordinate system as the reconstructed surfaces, meaning that the calculated distances would change if the object was moved or rotated. Three distance sensors shown in Figure 2.13 were used for testing the proposed solution - a ultrasonic distance sensor SR04, a low cost LiDAR rpLiDAR-A1 and a highly specialized professional LiDAR - the Hokuyo UTM-30LX. These sensors were mounted on the camera (Figure 2.18) and for each taken image for the SfM reconstruction their distance readings to the surface of the object were captured. Later on when both the surface and the camera positions were reconstructed, a synthetic distance readings were made from the calibrated position where the sensor was compared to the camera. The real world and synthetic readings were compared and from their ratio the absolute scale of the reconstruction was calculated (Figure 2.19).

The same covariance propagation approach was used to calculate the uncertainty of the resultant scale. It was shown that because of the lower uncertainty of the distance sensors, the calculated scale had greater precision

## References



**Fig. 2.19:** Finding the absolute scale of the reconstructed object, using the ratio between real world captured distances and the synthetic distances measured from the SfM object and cameras [51], paper F

than the GPS positioning approach. In addition, it was demonstrated that even the low cost distance sensor could provide highly precise scale calculations, while the more expensive LiDAR sensors were more robust against imperfections in the reconstructed surfaces like noise or holes.

With this the data capturing and correction parts of the thesis are covered, the next chapter tackles the problems of SfM 3D data analysis for noise detection, as well as for use in different applications, such as surface inspection and quantifying and capturing surface roughness.

## References

- [1] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer, "A survey of structure from motion\*." *Acta Numerica*, vol. 26, pp. 305–364, 2017.
- [2] T. Luhmann, S. Robson, S. Kyle, and J. Boehm, *Close-range photogrammetry and 3D imaging*. Walter de Gruyter, 2013.
- [3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [4] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.
- [5] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *International workshop on vision algorithms*. Springer, 1999, pp. 298–372.
- [6] F. Javadnejad, M. Gillins, and D. Gillins, "Vertical accuracy assessment of image-based reconstructed 3d point clouds with respect to horizontal ground sampling distance," in *ASPRS Annual Conference, Imaging & Geospatial Technology Forum (IGTF) 2016*, 2016.

## References

- [7] B. Peterson, *Understanding exposure: how to shoot great photographs with any camera*. AmPhoto books, 2016.
- [8] D. A. Kerr, "Apex-additive system of photographic exposure," *Issue*, vol. 7, no. 2007.08, p. 04, 2007.
- [9] S. Wanner and B. Goldluecke, "Reconstructing reflective and transparent surfaces from epipolar plane images," in *German Conference on Pattern Recognition*. Springer, 2013, pp. 1–10.
- [10] A. H. Ahmadabadian, A. Karami, and R. Yazdan, "An automatic 3d reconstruction system for texture-less objects," *Robotics and Autonomous Systems*, vol. 117, pp. 29–39, 2019.
- [11] S. Gosh, "History of photogrammetry," *Laval University, Canada*, 1981.
- [12] J. Albertz, "A look back," *Photogrammetric Engineering & Remote Sensing*, vol. 73, no. 5, pp. 504–506, 2007.
- [13] K. Atkinson, "A review of close-range engineering photogrammetry," *Photogrammetric Engineering and Remote Sensing*, vol. 42, no. 1, pp. 57–69, 1976.
- [14] R. Jiang, D. V. Jáuregui, and K. R. White, "Close-range photogrammetry applications in bridge measurement: Literature review," *Measurement*, vol. 41, no. 8, pp. 823–834, 2008.
- [15] T. Schenk, "Introduction to photogrammetry," *The Ohio State University, Columbus*, vol. 106, 2005.
- [16] H. Mitchell and I. Newton, "Medical photogrammetric measurement: overview and prospects," *ISPRS journal of photogrammetry and remote sensing*, vol. 56, no. 5-6, pp. 286–294, 2002.
- [17] C. Fraser, H. Hanley, S. Cronk *et al.*, "Close-range photogrammetry for accident reconstruction," *Optical 3D Measurements VII*, (Gruen, A./Kahmen, H, Eds), vol. 2, pp. 115–123, 2005.
- [18] T. Mikita, P. Janata, and P. Surový, "Forest stand inventory based on combined aerial and terrestrial close-range photogrammetry," *Forests*, vol. 7, no. 8, p. 165, 2016.
- [19] T. Luhmann, "Close range photogrammetry for industrial applications," *ISPRS journal of photogrammetry and remote sensing*, vol. 65, no. 6, pp. 558–569, 2010.
- [20] N. Statham, "Use of photogrammetry in video games: a historical overview," *Games and Culture*, vol. 15, no. 3, pp. 289–307, 2020.

## References

- [21] C. W. VisualSFM, "A visual structure from motion system," 2011.
- [22] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3d," in *ACM Siggraph 2006 Papers*, 2006, pp. 835–846.
- [23] D. Cernea, "OpenMVS: Multi-view stereo reconstruction library," 2020. [Online]. Available: <https://cdcseacave.github.io/openMVS>
- [24] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 8, pp. 1362–1376, 2009.
- [25] Agisoft, "Metashape (photoscan)," <http://www.agisoft.com/>, 2010, accessed: 2020-06-15.
- [26] Bentley, "Contextcapture," <https://www.bentley.com/en/products/brands/contextcapture>, 2016, accessed: 2020-06-15.
- [27] CapturingReality, "Reality capture," <https://www.capturingreality.com/>, 2016, accessed: 2020-06-15.
- [28] 3dFlow, "3df zephyr," <http://www.3dflow.net/>, 2011, accessed: 2020-06-15.
- [29] E. Rupnik, M. Daakir, and M. P. Deseilligny, "Micmac—a free, open-source solution for photogrammetry," *Open Geospatial Data, Software and Standards*, vol. 2, no. 1, pp. 1–9, 2017.
- [30] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [31] AliceVision, "Meshroom: A 3D reconstruction software." 2018. [Online]. Available: <https://github.com/alicevision/meshroom>
- [32] T. P. Kersten, D. Omelanowsky, and M. Lindstaedt, "Investigations of low-cost systems for 3d reconstruction of small objects," in *Euro-Mediterranean Conference*. Springer, 2016, pp. 521–532.
- [33] F. Nex, F. Remondino, M. Gerke, H.-J. Przybilla, M. Baumker, and A. Zurhorst, "Isprs benchmark for multi-platform photogrammetry." *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 2, 2015.
- [34] A. Koutsoudis, B. Vidmar, G. Ioannakis, F. Arnaoutoglou, G. Pavlidis, and C. Chamzas, "Multi-image 3d reconstruction data evaluation," *Journal of Cultural Heritage*, vol. 15, no. 1, pp. 73–79, 2014.

## References

- [35] A. Corradetti, K. McCaffrey, N. De Paola, and S. Tavani, "Evaluating roughness scaling properties of natural active fault surfaces by means of multi-view photogrammetry," *Tectonophysics*, vol. 717, pp. 599–606, 2017.
- [36] J. Schöning and G. Heidemann, "Taxonomy of 3d sensors," *Argos*, vol. 3, no. P100, pp. 9–10, 2016.
- [37] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, p. 78, 2017.
- [38] D. Marelli, S. Bianco, L. Celona, and G. Ciocca, "A blender plug-in for comparing structure from motion pipelines," in *2018 IEEE 8th International Conference on Consumer Electronics-Berlin (ICCE-Berlin)*. IEEE, 2018, pp. 1–5.
- [39] S. Bianco, G. Ciocca, and D. Marelli, "Evaluating the performance of structure from motion pipelines," *Journal of Imaging*, vol. 4, no. 8, p. 98, 2018.
- [40] I. Nikolov and C. Madsen, "Benchmarking close-range structure from motion 3d reconstruction software under varying capturing conditions," in *Euro-Mediterranean Conference*. Springer, 2016, pp. 15–26.
- [41] —, "Ggg-benchmarksfm: Dataset for benchmarking close-range sfm software performance under varying capturing conditions," 2020. [Online]. Available: <https://data.mendeley.com/datasets/bzxxk2n78s9/1>
- [42] I. Nikolov, "Ggg-benchmarksfm- secondary: Secondary dataset for benchmarking close-range sfm software performance captured with different cameras," 2020. [Online]. Available: <https://data.mendeley.com/datasets/t4d8mv3fxt/1>
- [43] I. Nikolov and C. B. Madsen, "Interactive environment for testing sfm image capture configurations." in *VISIGRAPP (1: GRAPP)*, 2019, pp. 317–322.
- [44] I. Nikolov and C. Madsen, "Lidar-based 2d localization and mapping system using elliptical distance correction models for uav wind turbine blade inspection." in *VISIGRAPP (6: VISAPP)*, 2017, pp. 418–425.
- [45] B. E. Schäfer, D. Picchi, T. Engelhardt, and D. Abel, "Multicopter unmanned aerial vehicle for automated inspection of wind turbines," in *2016 24th Mediterranean Conference on Control and Automation (MED)*. IEEE, 2016, pp. 244–249.

## References

- [46] R. Zlot and M. Bosse, "Efficient large-scale 3d mobile mapping and surface reconstruction of an underground mine," in *Field and service robotics*. Springer, 2014, pp. 479–493.
- [47] C. Galleguillos, A. Zorrilla, A. Jimenez, L. Diaz, Á. Montiano, M. Barroso, A. Viguria, and F. Lasagni, "Thermographic non-destructive inspection of wind turbine blades using unmanned aerial systems," *Plastics, Rubber and Composites*, 2015.
- [48] M. Cordero, M. Trujillo, J. Ruiz, A. Jimenez, L. Diaz, and A. Viguria, "Flexible framework for the development of versatile mav systems for multi-disciplinary applications," in *IMAV 2014: International Micro Air Vehicle Conference and Competition 2014, Delft, The Netherlands, August 12-15, 2014*. Delft University of Technology, 2014.
- [49] I. Nikolov, E. Kruse, and C. Madsen, "How capturing setups influence the quality of sfm reconstructions for wind turbine blade surface inspection," accepted at SPIE Future Sensing Technologies.
- [50] I. Nikolov and C. Madsen, "Performance characterization of absolute scale computation for 3d structure from motion reconstruction," in *14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (Visigrapp 2019) International Conference on Computer Vision Theory and Applications*. SCITEPRESS Digital Library, 2019, pp. 884–891.
- [51] I. Nikolov and C. B. Madsen, "Calculating absolute scale and scale uncertainty for sfm using distance sensor measurements: A lightweight and flexible approach," in *Recent Advances in 3D Imaging, Modeling, and Reconstruction*. IGI Global, 2020, pp. 168–192.
- [52] A. Lucieer, S. M. d. Jong, and D. Turner, "Mapping landslide displacements using structure from motion (sfm) and image correlation of multi-temporal uav photography," *Progress in Physical Geography*, vol. 38, no. 1, pp. 97–116, 2014.
- [53] M. Rumpler, S. Daftry, A. Tscharf, R. Pretenthaler, C. Hoppe, G. Mayer, and H. Bischof, "Automated end-to-end workflow for precise and geo-accurate reconstructions using fiducial markers," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 3, pp. 135–142, 2014.
- [54] L. Li and H. Lan, "Recovering absolute scale for structure from motion using the law of free fall," *Optics & Laser Technology*, vol. 112, pp. 514–523, 2019.

## References

- [55] D. Zhou, Y. Dai, and H. Li, "Reliable scale estimation and correction for monocular visual odometry," in *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2016, pp. 490–495.
- [56] R. Giubilato, S. Chiodini, M. Pertile, and S. Debei, "Scale correct monocular visual odometry using a lidar altimeter," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3694–3700.
- [57] M. Rabah, M. Basiouny, E. Ghanem, and A. Elhadary, "Using rtk and vrs in direct geo-referencing of the uav imagery," *NRIAG Journal of Astronomy and Geophysics*, 2018.
- [58] O. Kim and D.-J. Kang, "A sensor fusion method to solve the scale ambiguity of single image by combining imu," in *2015 15th International Conference on Control, Automation and Systems (ICCAS)*. IEEE, 2015, pp. 923–925.
- [59] T. Schöps, T. Sattler, C. Häne, and M. Pollefeys, "3d modeling on the go: Interactive 3d reconstruction of large-scale scenes on mobile devices," in *3D Vision (3DV), 2015 International Conference on*. IEEE, 2015, pp. 291–299.
- [60] L. Ding and G. Sharma, "Fusing structure from motion and lidar for dense accurate depth map estimation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 1283–1287.
- [61] M. Sarker, T. Ali, A. Abdelfatah, S. Yehia, and A. Elaksher, "a cost-effective method for crack detection and measurement on concrete surface," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, p. 237, 2017.
- [62] A. Corti, S. Giancola, G. Mainetti, and R. Sala, "A metrological characterization of the kinect v2 time-of-flight camera," *Robotics and Autonomous Systems*, vol. 75, pp. 584–594, 2016.
- [63] S. Du, N. Zheng, S. Ying, and J. Liu, "Affine iterative closest point algorithm for point set registration," *Pattern Recognition Letters*, vol. 31, no. 9, pp. 791–799, 2010.
- [64] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 4, pp. 376–380, 1991.
- [65] R. M. Haralick, "Propagating covariance in computer vision," in *Performance Characterization in Computer Vision*. Springer, 2000, pp. 95–114.

## References

- [66] C. B. Madsen, "A comparative study of the robustness of two pose estimation techniques," *Machine Vision and Applications*, vol. 9, no. 5-6, pp. 291–303, 1997.
- [67] I. Nikolov and C. Madsen, "Ggg-positioningsfmscale: Dataset for testing scaling and uncertainty propagation through camera positioning data," 2020. [Online]. Available: <https://data.mendeley.com/datasets/24vt4rbpyx/1>

## Chapter 3

# Structure from Motion Data Analysis

This chapter of the thesis continues the work presented in Chapter 2, by looking at ways to analyse and use the 3D data obtained from the SfM reconstructions, after they have been scaled to an absolute real world scale. One of the most prominent use cases for 3D reconstructions is for surface inspection in the industries. As part of the larger research project of wind turbine blade edge inspection, a particular interest presents the quality of surface information that can be extracted from an SfM reconstructions and how it matches to more traditional means like microscopy analysis. Of particular focus to this chapter is also how can surface roughness be separated from noise cause by imperfect 3D reconstructions. This is important, because small details can be lost if surface noise is prevalent on the reconstructions. Finally, the quantification and separation of different scales of surface roughness is also of interest, so a more ordered classification of different surface can be made for either inspection purposes or for transference to other mediums. Some of the text present in this chapter is an edited version of parts from the papers included in the thesis [1], as well as the in processing papers [2–4] (ADD NOT PUBLISHED PAPERS HERE).

### 3.1 Reconstruction Quality Analysis

#### 3.1.1 Introduction

The analysis of the quality of SfM reconstruction is an important transitional part between capturing the data and using it to extract and quantify information from the real world objects and surfaces. The quality analysis can be

separated into two parts - separation of signal and noise captured from the SfM and the quality of the captured signal itself.

The separation between signal and noise for point clouds and 3D meshes can be a non-trivial problem, as the vision of the two can become subjective depending on the use case and the analysed surface. In some cases it becomes much easier to surface smoothing algorithms, which remove the noise, but can also destroy small geometrical details in the process. Surface smoothing algorithms can be roughly divided into two types - global isotropic smoothing and local anisotropic smoothing. The first type is present in many of the SfM reconstruction pipelines, as well as most of the 3D modeling software solutions. These algorithms do not look at the underlying geometry, but provide a global smoothing, which makes them faster, especially on large and complex meshes. The drawback is that they tend to smooth out features and if larger smoothing values are used, they can even change the overall shape and size of the mesh. Examples of such isotropic smoothing algorithms can be seen in the research by [5–7]. On the other hand anisotropic algorithms, tend to look at the underlying local geometry of the meshes to try to tweak the severity of the smoothing and preserve smaller details. These algorithms use statistical methods [8] or machine learning type descriptors [9, 10] to analyse the underlying structure of the mesh.

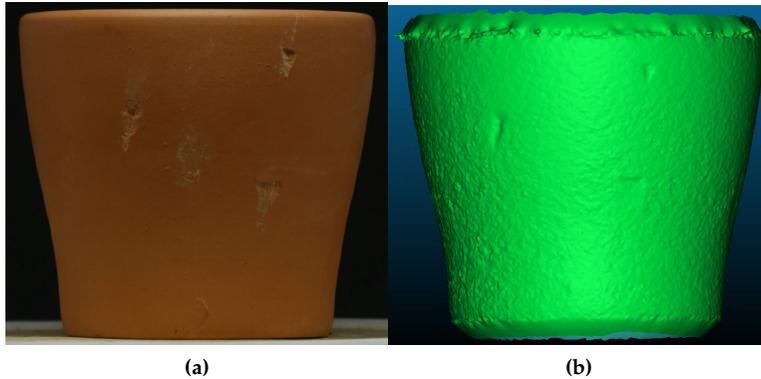


**Fig. 3.1:** Reconstructed mesh surface before (Figure 3.1a) and after (Figure 3.1b) a global smoothing algorithm is applied to remove the noisy parts. It can be seen how much of the real roughness is removed in the process

These approaches work for creating aesthetically pleasing reconstructions, where smaller details are captured by the texture maps and not by the mesh geometry. These results are good enough for cultural heritage, architectural or archaeological use cases, where sub-millimeter accuracy is not always important or for geological use cases, where the larger details are of more interest, than the smaller ones. This becomes a problem, when the surface reconstructions need to represent sub-millimeter or lower details and when the size, shape and depth of the surface roughness is required. An example

### 3.1. Reconstruction Quality Analysis

of this is given in Figure 3.1, where a part of a wind turbine blade's surface is shown before and after a global smoothing algorithm. It can easily be seen that the smaller details are removed and how much of the information is lost in the process.



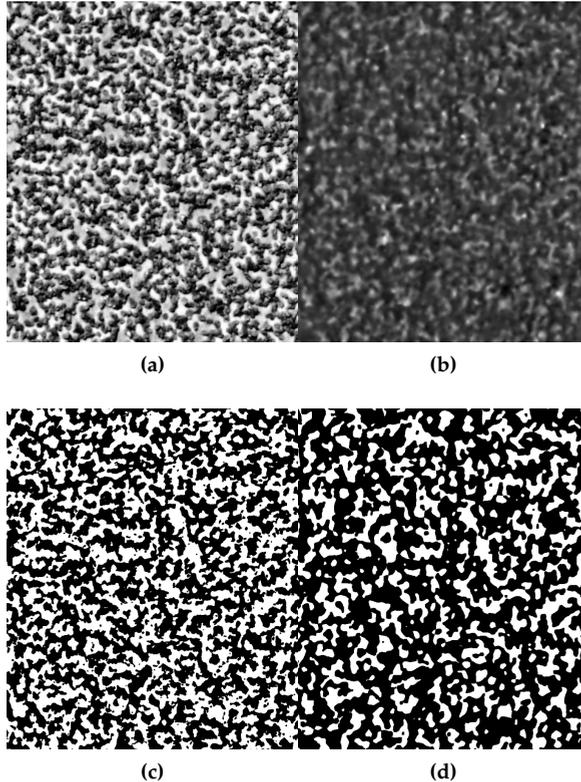
**Fig. 3.2:** A real life cup and the reconstruction captured under sub-optimal conditions. The surface of the reconstruction has a number of noisy patches and geometrical errors and does not represent the real roughness correctly.

In the case for surface inspection, it is better to be able to identify the noisy surface patches, so they can be removed from the calculation of the overall roughness metrics. The necessity for removing patches is demonstrated in Figure 3.2, where the smooth surface of a cup, has been reconstructed, under sub-optimal capturing conditions, resulting in a rough mesh, which does not represent the real life physical characteristics of the imaged object. A number of challenges make noise identification a non-trivial problem:

- Surfaces can have varying roughness, with abrupt changes between very smooth and coarse patches
- The shape of the surface can have abrupt changes, ridges and holes
- For the noise identification to be generalized enough, it needs to be able to detect noise on objects with different sizes

The second part of the quality analysis, after the noise has been separated from the roughness, looks to evaluate the captured surface geometry and to determine if it can be used instead of traditional surface analysis methods. Normally contact measurements [11], computer vision methods [12, 13] or microscopy [14, 15] are used for capturing surface roughness and irregularities in the industries. For SfM reconstruction to be useful on the same scale, the details captured need to be verified. An example for this can be seen in

Figure 3.3, where a roughness patch of sandpaper has been reconstructed using SfM and the resultant mesh has been rasterized to capture the depth map seen from the same direction. In Figure 3.3c and 3.3d, both images have been segmented to capture only the sand grains representing the roughness. The SfM reconstruction has captured the overall structure, but much of it has also been lost or changed in the process. The severity of the loss of information, needs to be quantified, before using the SfM reconstructions further on.



**Fig. 3.3:** Example of possible information loss, when reconstructing a 3D surface. A sandpaper patch (Figure 3.3a), has been reconstructed and the depth map (3.3b) has been created. When segmenting the sand grit from each of the two (Figure 3.3c and 3.3d), its shape and structure is changed.

### 3.1.2 State of the Art

The separation between surface information and noise has been a widely research field in computer vision. The most straightforward way for detecting noise and geometrical errors on 3D surfaces is by having ground truth of the



describing the overall curvature of local patches on the surface and using the normal information to detect changes in the roughness. The papers by [16] and [22] extend that research by introducing local features and more discriminative models, which can better detect changes on the edges and seams of models.

Other methods build even further on these approaches by fitting geometrical shapes onto the curvature of the 3D surface and calculating roughness ratios [19] (Figure 3.5) or by calculating the scale invariant and robust saliency measurements [23] and extracting local 2D and 3D features from neighbourhoods of points [24].

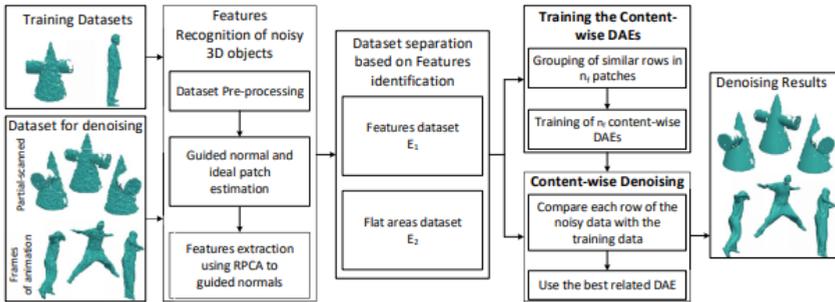


Fig. 3.6: Pipeline for deep learning mesh denoising, by detecting the features of noise on 3D meshes, by [25]

Finally, the manual extraction of features can be completely subverted, if enough data is present by the use of deep learning methods for surface evaluation and denoising [25–27]. These methods normally produce better results, but require large amounts of data (Figure 3.6), no always present for different types of objects and surfaces.

### 3.1.3 Contributions

As part of the thesis, two papers (paper G and paper H) [2, 3] have been written on the topic of quality assessment of SfM reconstructions. The first paper "*Rough or Noisy? Metrics for Noise Estimation in SfM Reconstructions* [2]" (given as paper G) is directed towards finding a generic way for separating reconstruction noise from surface roughness information. The paper extends the normally used surface geometry metrics for roughness and noise detection, by combining them with ones extracted from SfM camera positions and the capturing environment, to try to determine which parts of the mesh are most likely to contain noise. Three general mesh based metrics are chosen for this - local roughness from Gaussian curvature [21], difference of normal [28] and vertex local spacial density [29], together with a metric extracted from

### 3.1. Reconstruction Quality Analysis

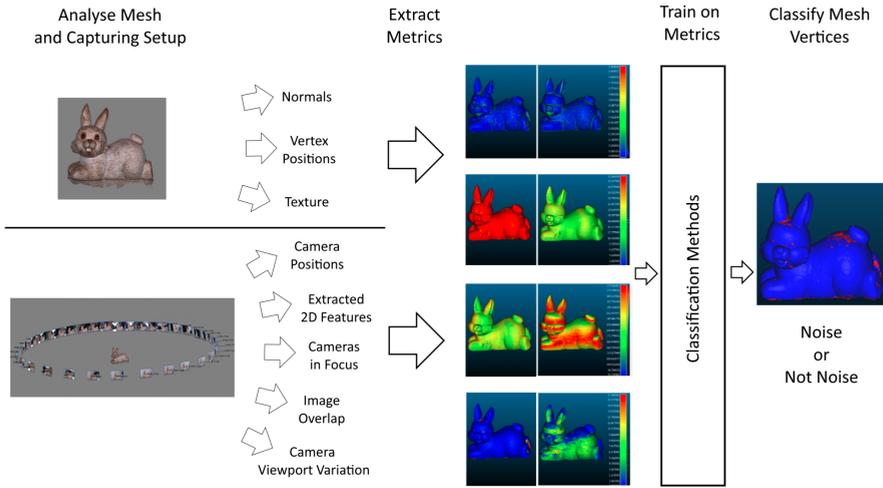
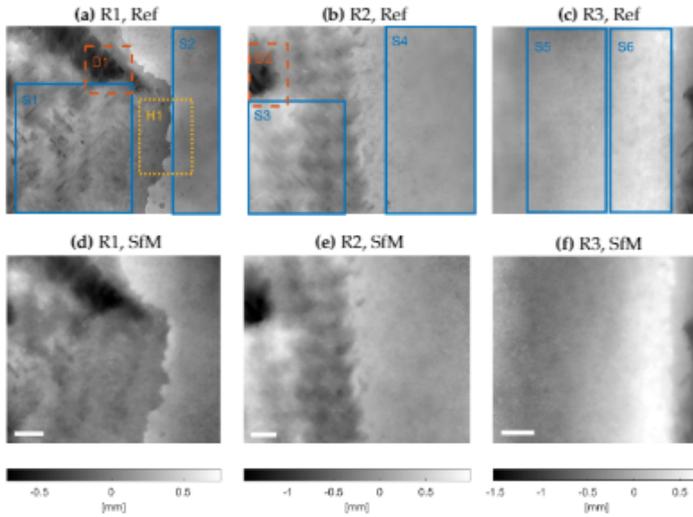


Fig. 3.7: The proposed metrics for estimating noise on SfM reconstructions [2]

the reconstructed mesh texture using the local color entropy [30].

These generic metrics are combined with ones specific to SfM reconstructions, which use the additional capturing information that comes from using the algorithm. The specific SfM metrics, are selected after analysing the requirements for successful reconstructions, demonstrated by benchmarking and analysis research. These metrics are as follows - number of cameras seeing each vertex [31], projected 2D features [32], vertices in focus [33], vertices seen from parallel to the surface cameras [34], vertex area of visibility [34]. These metrics are then used for training an Adaptive Boosting ensemble classifier for estimating noisy from non-noisy surface vertices (Figure 3.7). For this objects with different shapes and sizes, were reconstructed and manually annotated into noisy and clean parts and used as training data. We demonstrated that the algorithms can be used for discriminating between noisy and clear surfaces, even when surfaces were rough and damaged. Three main conclusions are taken from the work on this paper:

- An automatic separation of noise and roughness can be achieved with an accuracy above 85% and used as an initial step, before analysing reconstructed surfaces. The results from the paper show that in some cases additional manual inspection would be needed.
- Utilizing SfM image capturing information for estimating the presence of possible noise, can help with separation of noise from rough surfaces and produces better results than just using geometrical features, which adds another 5-6% accuracy to the noise estimation model.



**Fig. 3.8:** Comparison between different patches of a wind turbine blade surface reconstructed using SfM and confocal microscopy [3], paper H

- The selected features can be used in a larger context, with a classifier trained on generic objects, being used on special surfaces like wind turbine blades without any noticeable accuracy loss and with precision of 79% and recall of 88% .

The second paper "*High-resolution Structure-from-Motion for quantitative measurement of leading edge roughness* [3]" (presented as paper H) focuses on verifying the quality of the SfM reconstruction and determining if it can be used for sub-millimeter surface inspections of wind turbine blades. To test this a wind turbine blade is reconstructed both using a SfM algorithm and using elastomer replica material microscopy [15, 35]. A number of patches of the surface (Figure 3.8) are then compared using ISO 25178 standard surface roughness parameters [36] - root mean square height  $S_q$ , root mean square of the surface gradient  $S_{dq}$  and spacial distance at which the texture statistically different  $S_{dl}$ . The paper showed that using a high resolution camera and optimal capturing settings and conditions, SfM can achieve sub-millimeter performance and can capture details close to those from confocal microscopy. The comparison between the microscopy measurements and SfM has led to two conclusions:

- SfM can be used to capture surface roughness with sub-millimeter accuracy, provided that a high enough resolution camera is used and good capturing conditions are provided.

## 3.2. Reconstructed Surface Classification

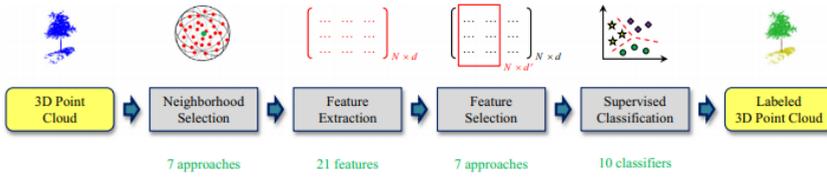


Fig. 3.9: Point cloud semantic segmentation based on extracted 2D and 3D features and using classical statistical classifiers [24]

- SfM reconstructions provide a somewhat blurrier representation of the surface, than the combination of replica moulding and confocal microscopy, but the speed of SfM and the ease of use make it non the less suitable for wind turbine blade inspection.

## 3.2 Reconstructed Surface Classification

Once a reconstructed mesh is cleaned up of noise and its quality is validated, it can be used for extracting information from the inspected surface or object. As part of the Leading Edge Roughness project, this data was used to determine how much the energy performance of a wind turbine blade would degrade, depending on the surface roughness of its blades. The algorithms used for these calculations could not be published, as they are confidential to some of the companies involved in the project. In addition to the main body of work shown in the previous sections, two use cases of SfM surface roughness data would be presented in this section, demonstrating both ways to quantify surface data and use it to create multimodal representation on the real world object.

### 3.2.1 State of the Art

Because 3D surface classification is a extensive field, for the state of the art of this section, only the research in the same direction as thesis study will be covered. SfM reconstructions can be useful for representing surface information for detecting micro and macro roughness, geometrical irregularities or points of interest. These can be later leveraged for things like mesh and point cloud segmentation and classification or damage detection. Some examples of that are semantic segmentation of roads [37], separation of unstructured point clouds into classes depending on their geometrical properties for geological use [38], urban planning [39], forestry [40], among others. Other possible uses for the derived point cloud roughness can be for road analysis [41] and for the analysis of Earth surface changes [42].

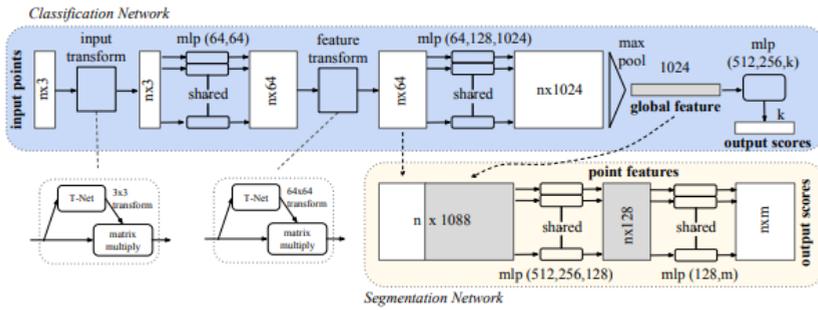


Fig. 3.10: PointNet - point cloud classification, part segmentation and semantic segmentation using convolutional neural networks [45]

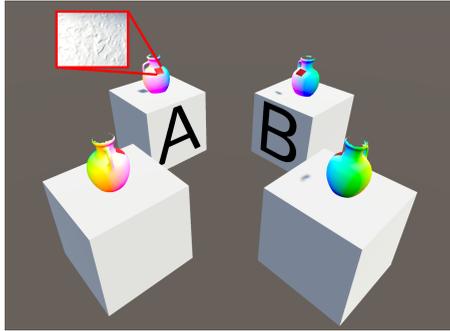
Roughness estimation and classification is mainly based on extracting 2D and 3D surface descriptors from the analysed point cloud or mesh and using them to train a supervised or semi-supervised classifiers. Those descriptors can range from 2D and 3D point cloud properties like point density, normal orientation and change, etc. [43], to principal component analysis (PCA) descriptors like linearity, planarity, anisotropy [24, 44] (Figure 3.9) and even learned descriptors from convolutional and deep neural networks [45, 46], if enough data for training is present (Figure 3.10).

Once the roughness and surface characteristics of a 3D object are extracted they can be used in a variety of cases. They can help with the better representation of simulations of fluid flow in real life surfaces [47], or give more information how roughness can influence the surfaces of roads [48] and on hill slopes [49], or even representing the surface roughness as haptic feedback for giving more information to professional dermatologists [50, 51].

### 3.2.2 Contributions

Two papers were written in the field of 3D surface classification [1, 4]. The two publications tackle different aspects of using the roughness data. The first paper "*Preliminary Study on the Use of Off-the-Shelf VR Controllers for Vibrotactile Differentiation of Levels of Roughness on Meshes*" [1] (Paper I in the thesis), focuses on how the roughness from SfM reconstructed objects be used for extracting haptic feedback for an immersive virtual reality experience. The paper presents a preliminary study on how the HTC Vive controllers' built-in vibro-tactile motors can be used to give information about the roughness of 3D meshes. The roughness is extracted using the difference of normals method [28] and used to separate the surface of the 3D object into two classes representing low and high roughness. These areas then influence the amplitude and frequency of the vibration on the controllers as users inter-

### 3.2. Reconstructed Surface Classification

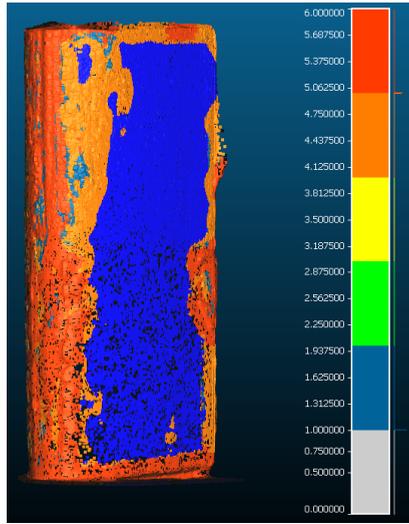


**Fig. 3.11:** Testing area for determining if users detect difference in roughness levels only by the vibrotactile sensation from the calculated roughness values [1], paper I

act with the 3D objects. The interaction is done through a virtual stylus and no real surface or objects are needed. It was demonstrated that the extracted roughness metrics can be used, as part of a vibro-tactile system to give users a better understanding of the 3D mesh, even when the mesh shape was the same and users could only use the vibration information to judge subtle differences (Figure 3.11). A number of conclusion are made from the results of the paper:

- Small scale roughness can be successfully captured using SfM reconstruction and used to represent the real surface.
- The amplitude and frequency of vibrations can be manipulated using the 3D mesh surface to create the sensation of "touching" surfaces with different roughness.
- More research into vibro-tactile actuators is needed to provide a better representation of the surface.

The second paper "*Quantifying Wind Turbine Blade Surfaces Using Sandpaper Grit Sizes* [4]" (paper J in the thesis) tries to use a comparison to the standardised roughness of sandpaper, to quantify the surface characteristics of wind turbine blades. Using sandpaper roughness to represent different wind turbine profiles has been widely used in the literature ( [52], [53]) and in the proposed paper, this idea is extended by using these easily accessible sandpaper surfaces as units of measurement for the blade surfaces. A number of sandpaper grit sizes from P40 to P240 [54] are mounted on foam blade replicas and reconstructed using SfM. Surface features are then extracted from the reconstructions like linearity, planarity, sphericity, change of curvature and others [24, 44]. These features are used to train a random forest statistical classification method, which is then used on wind turbine blade surfaces



**Fig. 3.12:** Example of wind turbine blade surface segmentation, based on the sandpaper roughness [4], paper J

containing damage and varying degrees of roughness. The blade is then segmented depending on which sandpaper grit size best describes its surface at each point, as seen in Figure 3.12. Based on the finding of the paper a number of conclusions can be made:

- The same way sandpapers with different grit sizes are used to replicate roughness on wind turbine blades, in wind tunnel tests, they can be used to classify blade surface roughness on 3D reconstructions.
- A combination of local area geometrical, statistical and covariance features can be used to explain the geometrical features of reconstructed surfaces.
- A classifier can be trained to separate wind turbine blade surfaces of different sizes into the proposed sandpaper grit sizes.

More research is needed to verify that the classified roughness areas, would behave the same way as a sandpaper of equivalent grit size under working conditions and this paper sets the stage for feature research like this.

## References

- [1] I. Nikolov, J. S. Høngaard, M. Kraus, and C. B. Madsen, "Preliminary study on the use of off-the-shelf vr controllers for vibrotactile differentiation of levels of roughness on meshes," in *VISIGRAPP (1: GRAPP)*, 2020, pp. 334–340.
- [2] I. Nikolov and C. Madsen, "Rough or noisy? metrics for noise detection in sfm reconstructions," accepted at MDPI Sensors Journal.
- [3] M. S. Nielsen, I. Nikolov, E. K. Kruse, J. Garnæs, and C. B. Madsen, "High-resolution structure-from-motion for quantitative measurement of leading-edge roughness," *Energies*, vol. 13, no. 15, p. 3916, 2020.
- [4] I. Nikolov and C. Madsen, "Quantifying wind turbine blade surfaces using sandpaper grit sizes," submitted.
- [5] B. Kim and J. Rossignac, "Geofilter: Geometric selection of mesh filter parameters," in *Computer Graphics Forum*, vol. 24, no. 3. Wiley Online Library, 2005, pp. 295–302.
- [6] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa, "Laplacian mesh optimization," in *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*. ACM, 2006, pp. 381–389.
- [7] Z.-X. Su, H. Wang, and J.-J. Cao, "Mesh denoising based on differential coordinates," in *2009 IEEE International Conference on Shape Modeling and Applications*. IEEE, 2009, pp. 1–6.
- [8] X. Lu, W. Chen, and S. Schaefer, "Robust mesh denoising via vertex pre-filtering and l1-median normal filtering," *Computer Aided Geometric Design*, vol. 54, pp. 49–60, 2017.
- [9] P.-S. Wang, Y. Liu, and X. Tong, "Mesh denoising via cascaded normal regression." *ACM Trans. Graph.*, vol. 35, no. 6, pp. 232–1, 2016.
- [10] X. Lu, X. Liu, Z. Deng, and W. Chen, "An efficient approach for feature-preserving mesh denoising," *Optics and Lasers in Engineering*, vol. 90, pp. 186–195, 2017.
- [11] C. Y. Poon and B. Bhushan, "Comparison of surface roughness measurements by stylus profiler, afm and non-contact optical profiler," *Wear*, vol. 190, no. 1, pp. 76–88, 1995.
- [12] O. B. Abouelatta, "3d surface roughness measurement using a light sectioning vision system," in *Proceedings of the world congress on engineering*, vol. 1, 2010, pp. 698–703.

## References

- [13] E. Gadelmawla, "Estimation of surface roughness for turning operations using image texture features," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 225, no. 8, pp. 1281–1292, 2011.
- [14] K. A. Alshibli and M. I. Alsaleh, "Characterizing surface roughness and shape of sands using digital microscopy," *Journal of computing in civil engineering*, vol. 18, no. 1, pp. 36–45, 2004.
- [15] H. Hansen, R. Hocken, and G. Tosello, "Replication of micro and nano surface geometries," *CIRP Annals - Manufacturing Technology*, vol. 60, no. 2, pp. 695–714, jan 2011. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0007850611002101http://www.sciencedirect.com/science/article/pii/S0007850611002101?via%3Dihub%3D>
- [16] R. Song, Y. Liu, R. R. Martin, and P. L. Rosin, "Mesh saliency via spectral processing," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 1, p. 6, 2014.
- [17] V. Knyaz and S. Zheltov, "Accuracy evaluation of structure from motion surface 3d reconstruction," in *Videometrics, Range Imaging, and Applications XIV*, vol. 10332. International Society for Optics and Photonics, 2017, p. 103320P.
- [18] Ž. Santoši, I. Budak, V. Stojaković, M. Šokac, and Đ. Vukelić, "Evaluation of synthetically generated patterns for image-based 3d reconstruction of texture-less objects," *Measurement*, vol. 147, p. 106883, 2019.
- [19] P. Lai, C. Samson, and P. Bose, "Surface roughness of rock faces through the curvature of triangulated meshes," *Computers & Geosciences*, vol. 70, pp. 229–237, 2014.
- [20] G. Lavoué, "A local roughness measure for 3d meshes and its application to visual masking," *ACM Transactions on Applied perception (TAP)*, vol. 5, no. 4, p. 21, 2009.
- [21] K. Wang, F. Torkhani, and A. Montanvert, "A fast roughness-based approach to the assessment of 3d mesh visual quality," *Computers & Graphics*, vol. 36, no. 7, pp. 808–818, 2012.
- [22] J. Guo, V. Vidal, A. Baskurt, and G. Lavoué, "Evaluating the local visibility of geometric artifacts," in *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception*. ACM, 2015, pp. 91–98.

## References

- [23] A. El Chakik, A. R. El Sayed, and S. Nohra, "An invariant multi-scale saliency detection for 3d mesh," in *2018 4th International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, 2018, pp. 310–314.
- [24] M. Weinmann, B. Jutzi, S. Hinz, and C. Mallet, "Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 105, pp. 286–304, 2015.
- [25] G. Arvanitis, A. Lalos, and K. Moustakas, "Feature-aware and content-wise denoising of 3d static and dynamic meshes using deep autoencoders," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2019, pp. 97–102.
- [26] I. Abouelaziz, M. El Hassouni, and H. Cherifi, "A convolutional neural network framework for blind mesh visual quality assessment," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 755–759.
- [27] C. Duan, S. Chen, and J. Kovacevic, "3d point cloud denoising via deep neural network based local surface estimation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8553–8557.
- [28] Y. Ioannou, B. Taati, R. Harrap, and M. Greenspan, "Difference of normals as a multi-scale operator in unorganized point clouds," in *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*. IEEE, 2012, pp. 501–508.
- [29] T. Rabbani, F. Van Den Heuvel, and G. Vosselmann, "Segmentation of point clouds using smoothness constraint," *International archives of photogrammetry, remote sensing and spatial information sciences*, vol. 36, no. 5, pp. 248–253, 2006.
- [30] S. Harwin and A. Lucieer, "Assessing the accuracy of georeferenced point clouds produced via multi-view stereopsis from unmanned aerial vehicle (uav) imagery," *Remote Sensing*, vol. 4, no. 6, pp. 1573–1599, 2012.
- [31] S. W. Hixon, C. P. Lipo, T. L. Hunt, and C. Lee, "Using structure from motion mapping to record and analyze details of the colossal hats (pukao) of monumental statues on rapa nui (easter island)," *Advances in Archaeological Practice*, vol. 6, no. 1, pp. 42–57, 2018.
- [32] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer, "A survey of structure from motion\*." *Acta Numerica*, vol. 26, pp. 305–364, 2017.

## References

- [33] D. Zhang, R. Watson, G. Dobie, C. MacLeod, A. Khan, and G. Pierce, "Quantifying impacts on remote photogrammetric inspection using unmanned aerial vehicles," *Engineering Structures*, p. 109940, 2020.
- [34] M. Marčiš, "Quality of 3d models generated by sfm technology," *Slovak Journal of Civil Engineering*, vol. 21, no. 4, pp. 13–24, 2013.
- [35] F. Baruffi, P. Parenti, F. Cacciatore, M. Annoni, and G. Tosello, "On the Application of Replica Molding Technology for the Indirect Measurement of Surface and Geometry of Micromilled Components," *Micromachines*, vol. 8, no. 6, p. 195, jun 2017. [Online]. Available: <http://www.mdpi.com/2072-666X/8/6/195>
- [36] "Iso 25178 geometrical product specifications (gps) – surface texture: Areal – part 2: Terms, definitions and surface texture parameters," <https://www.iso.org/standard/42785.html>, accessed: 2020-03-05.
- [37] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg, "Joint semantic segmentation and 3d reconstruction from monocular video," in *European Conference on Computer Vision*. Springer, 2014, pp. 703–718.
- [38] T. Dewez, D. Girardeau-Montaut, C. Allanic, and J. Rohmer, "Facets: A cloudcompare plugin to extract geological planes from unstructured 3d point clouds," 2016.
- [39] A. Martinovic, J. Knopp, H. Riemenschneider, and L. Van Gool, "3d all the way: Semantic segmentation of urban scenes from start to end in 3d," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4456–4465.
- [40] D. Fawcett, B. Azlan, T. C. Hill, L. K. Kho, J. Bennie, and K. Anderson, "Unmanned aerial vehicle (uav) derived structure-from-motion photogrammetry point clouds for oil palm (*elaeis guineensis*) canopy segmentation and height estimation," *International Journal of Remote Sensing*, vol. 40, no. 19, pp. 7538–7560, 2019.
- [41] A. A. Alhasan, K. Younkin, and D. J. White, "Comparison of roadway roughness derived from lidar and sfm 3d point clouds," 2015.
- [42] K. L. Cook, "An evaluation of the effectiveness of low-cost uavs and structure from motion for geomorphic change detection," *Geomorphology*, vol. 278, pp. 195–208, 2017.
- [43] N. Brodu and D. Lague, "3d terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 68, pp. 121–134, 2012.

## References

- [44] D. Yarotsky, "Geometric features for voxel-based surface recognition," *arXiv preprint arXiv:1701.04249*, 2017.
- [45] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.
- [46] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," in *Advances in neural information processing systems*, 2018, pp. 820–830.
- [47] M. Wang, Y.-F. Chen, G.-W. Ma, J.-Q. Zhou, and C.-B. Zhou, "Influence of surface roughness on nonlinear flow behaviors in 3d self-affine rough fractures: Lattice boltzmann simulations," *Advances in water resources*, vol. 96, pp. 373–388, 2016.
- [48] L. Díaz-Vilariño, H. González-Jorge, M. Bueno, P. Arias, and I. Puente, "Automatic classification of urban pavements using mobile lidar data and roughness descriptors," *Construction and Building Materials*, vol. 102, pp. 208–215, 2016.
- [49]
- [50] K. Kim, "Roughness based perceptual analysis towards digital skin imaging system with haptic feedback," *Skin Research and Technology*, vol. 22, no. 3, pp. 334–340, 2016.
- [51] —, "Image-based haptic roughness estimation and rendering for haptic palpation from in vivo skin image," *Medical & biological engineering & computing*, vol. 56, no. 3, pp. 413–420, 2018.
- [52] O. F. Marzuki, A. S. M. Rafie, F. I. Romli, and K. A. Ahmad, "Magnus wind turbine: the effect of sandpaper surface roughness on cylinder blades," *Acta Mechanica*, vol. 229, no. 1, pp. 71–85, 2018.
- [53] M. S. Genc, K. Kemal, and H. H. Açıkel, "Investigation of pre-stall flow control on wind turbine blade airfoil using roughness element," *Energy*, vol. 176, pp. 320–334, 2019.
- [54] "Iso 6344(1998) coated abrasives — grain size analysis parts 1-3," <https://www.iso.org/standard/12643.html>, accessed: 2019-03-06.

## References

# Chapter 4

## Additional Project Work

As part of the larger project the PhD was connected to, a number of activities and developments were carried out, which could not be documented, in the form of papers or publications, because of their proprietary nature, connected to the participating companies. In this chapter a quick overview will be given for these activities, which are deemed useful for better understanding how the papers written for the thesis tie to the larger project. Additional details will be omitted for the sake of brevity.

### 4.1 Comparison Tests between SfM and Microscopy

A number of testing scenarios for comparison between the outputs of SfM and confocal microscopy were evaluated, later culminating with the publication of two papers [1](paper H) and [2](paper J). The more significant of these testing scenarios are given in the subsections below.

#### 4.1.1 Testing Wind Turbine Surface Reconstructions

To verify the quality of the wind turbine blade reconstructions produced by different SfM solutions, their output was compared to that of a confocal microscopy. The initial tests were performed on decommissioned blades, stored indoors. A number of surface patches were selected, roughly separated into three categories - damaged areas, rough areas and clean areas. Examples of these patches can be seen in Figure 4.1. The captured images were reconstructed with the two SfM solutions, which performed best in the benchmark presented in paper A [3] - Agisoft Metashape [4] and Bentley ContextCapture [5]. These reconstructions were then compared to ground truth surface captures using a Hirox RH-2000 microscope [6], calculating the root mean

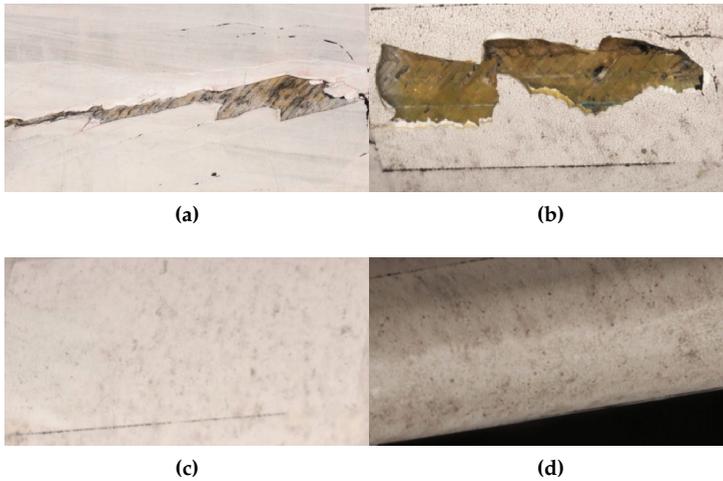


Fig. 4.1: Examples of blade surfaces tested as part of the initial indoor image capturing.

square error between them. An example of the reconstructions from the two programs and the microscopy can be seen in Figure 4.2.

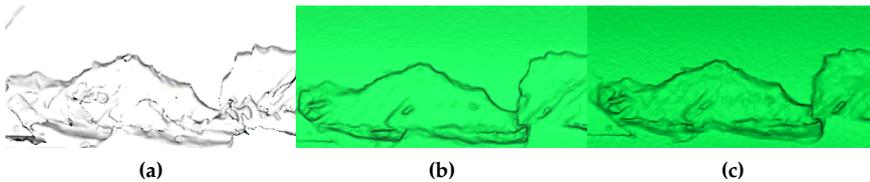


Fig. 4.2: Examples of comparison output from microscopy reconstruction (Figure 4.2a), Bentley ContextCapture SfM (Figure 4.2b) and Agisoft Metashape SfM (Figure 4.2c).

These experiments were later repeated on a smaller wind turbine blade segment, which was sandblasted to achieve a sufficient erosion level. A number of patches were again chosen representing the same three categories as with the previous experiment. For this experiment, the blade segment was imaged outdoor, under more complex lighting conditions. This was done to determine if the results obtained from the previous experiment, could be repeated outdoor and how much would the quality of the reconstruction suffer in the process.

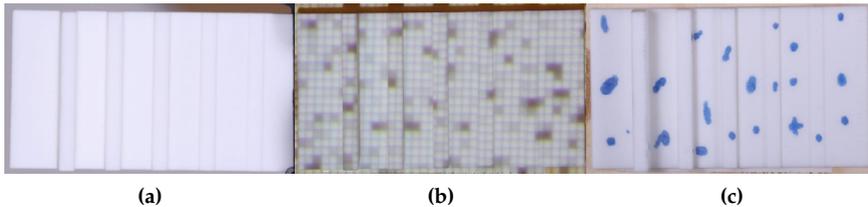
From these early experiments, it was determined that SfM could provide a sufficient reconstruction of the surface features, as long as good capturing conditions were present, but additional experiments were needed to demonstrate how close to a sub-millimeter accuracy the reconstruction could

#### 4.1. Comparison Tests between SfM and Microscopy

achieve. This was proved later in the paper H [1] in a setup, which more closely matched the real world one.

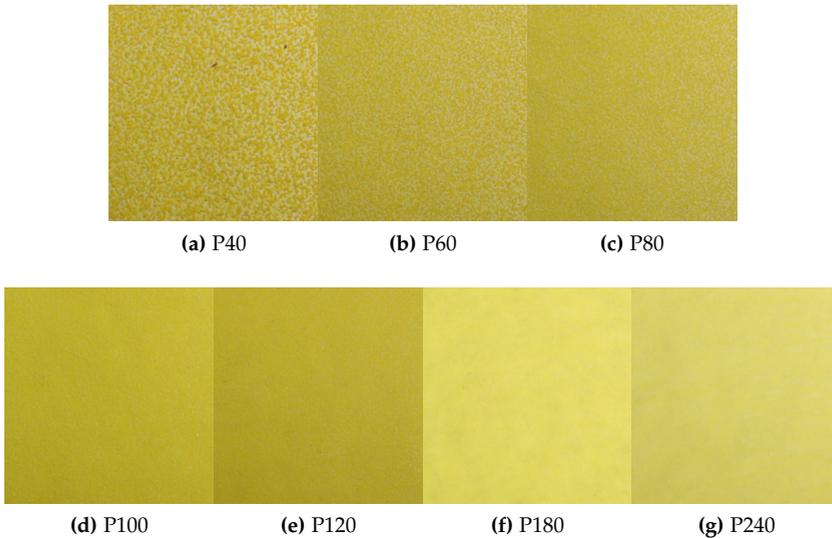
#### 4.1.2 Testing the Influence of Texture on SfM Reconstructions

One of the main questions, that was left unanswered from the initial wind turbine tests was how much did the reconstruction quality depend on the surface texture and how much on the surface geometry and shape. To answer it a number of testing scenarios were created using custom build step artifacts and sandpaper patches. The step artifacts, were designed so, they could demonstrate the reconstruction quality at a number of different scales from 10 mm to 0.625 mm. For their reconstruction, different surface textures were introduced - the monochrome normal one, a projected pattern one and a randomized drawn one (Figure 4.3), to test out how these would affect the quality of the reconstruction. It was demonstrated that a more random and pronounced texture would yield better results, when compared to a ground truth captured using a comfocal microscopy.



**Fig. 4.3:** The step artefact, together with different ways of introducing a surface texture to it - by projecting it and directly drawn on.

The second part of the experiment was done using sandpaper patches with different grit sizes from P40 up until P240. These patches were attached to CnC milled wind turbine blade replicas (Figure 4.4) and reconstructed both using the SfM and the comfocal microscopy. These reconstructions were compared on the basis of the size and shape of the average sandpaper grain sizes, as well as the power spectral density (PSD) analysis. The results showed that SfM reconstructions could deliver the same shape and size of the grains and to follow the approximate shape of the PSD curve of the microscopy reconstructions. The same sandpaper patches were later used for the paper J [2], where the lessons learned from the representation of the sandpaper grit sizes were applied for classifying the surface roughness of wind turbine blades.

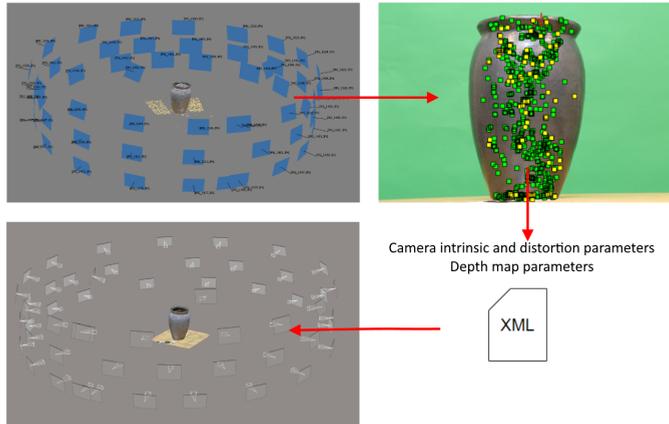


**Fig. 4.4:** The sandpaper grit sizes used for reconstruction and testing, the accuracy of SfM reconstructions, for capturing surfaces with different size roughness.

## 4.2 Combining SfM Solutions for Improving Reconstruction Quality

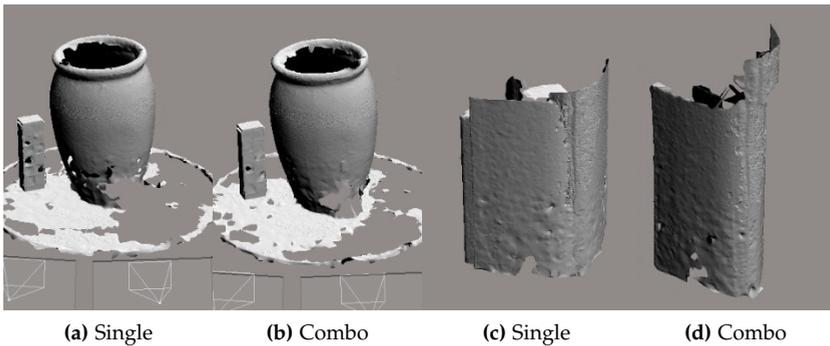
One of the lessons learning from creating the benchmarking paper [3] (paper A), was that SfM solutions can be roughly divided into two categories - those that produce high quality reconstructions, but are not robust to sub-optimal capturing conditions and those that produce less detailed reconstructions, but do not fail when not presented with optimal conditions. This prompted a research into combining the workflows of two solutions, each from one of the categories, to leverage their strengths. The two best performing applications from the benchmarking paper [3](Paper A), were chosen - Metashape and ContextCapture. Metashape (formally PhotoScan) was proven to achieve robust initial camera triangulation and sparse point clouds, but was unable to capture the smaller details of surfaces, with the required clarity. ContextCapture on the other hand was able to much easier achieve sub-millimeter accuracy, but it could also easily completely fail in the initial parts of the reconstruction pipeline, when the illumination and camera positions provided to it were not optimal. By combining the two applications (Figure 4.5), through a data parsing and rotation readjustment, cleaner and more robust results were achieved by removing holes and noise in the final reconstructed meshes (Figure 4.6). On the opposite side, the approach required more processing time

## 4.2. Combining SfM Solutions for Improving Reconstruction Quality



**Fig. 4.5:** Combining the two SfM solutions Metashape and ContextCapture into one processing pipeline

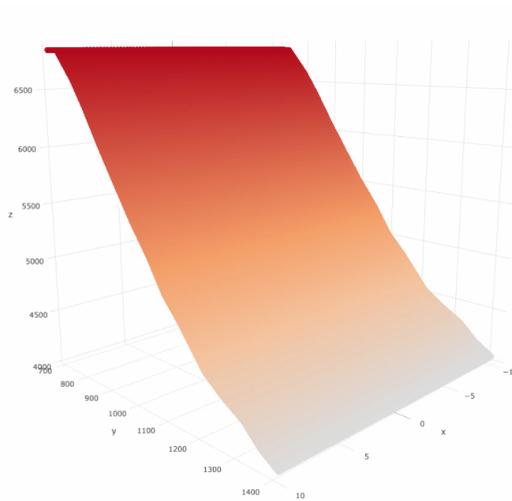
and the presence of the two SfM solutions, which made it harder to replicate the results in the long run.



**Fig. 4.6:** Difference between just using ContextCapture and a combo between MetaShape and ContextCapture

A publication was planned as part of the research done for this combination approach, but because of delays and work schedule changes, the final results were produced later in the project's lifetime. This together with the rapid development in commercial SfM applications, resulted in the results being comparable or worse than the current state of the art. It was decided not to pursue this research further.

### 4.3 LiDAR Intensity Roughness Detection



**Fig. 4.7:** Initial white paper calibration of the Hokuyo LiDAR - X axis shows the LiDAR ray angle, Y axis is the distance to the paper and Z axis is the intensity change

As part of the developed scanning drone platform and for calculating the absolute scale of the captured surface data in the paper F [7], a number of distance sensors were tested. One of these sensors is the Hokuyo LX-30LM [8] LiDAR. As part of the output data the sensor can calculate the intensity for each of the 360 degree readings. These intensities depend on the angle and distance at which they were taken, as well as the surface they were taken from. These readings have been used for determining the different materials in the environment [9] and for segmenting them, if necessary to help SLAM algorithms [10]. This was determined as a viable initial way for fast detection of areas of changed roughness on wind turbine blades. Once such areas were detected, the drone would perform a more detailed SfM scanning.

To be able to properly detect rougher or damaged areas on the wind turbine blade, the LiDAR's readings need to be calibrated, as the captured intensities vary with angle and distance. To be able to model these variations, it is suggested [10], that an initial calibration using a material with know or a stable reflectance is needed. A white sheet of paper can be such a surface. An initial calibration of the distance and angle variations of imaging a the sheet of paper is done (Figure 4.7) and is later used to calculate the relative reflectance of different surfaces, compared to it using the formulas presented by [10].

To test if the intensity changes can signalize changes in the roughness of a

### 4.3. LiDAR Intensity Roughness Detection

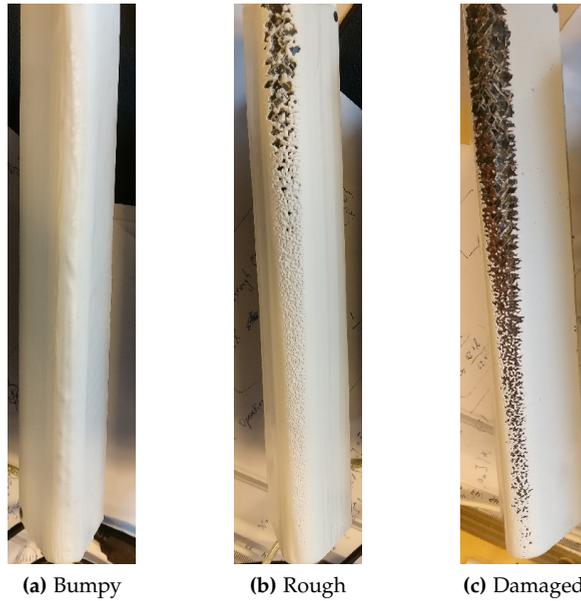


Fig. 4.8: Wind turbine blade pieces, with different degrees of surface damage

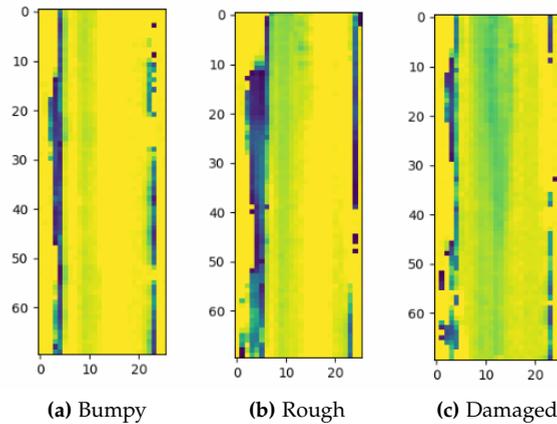


Fig. 4.9: Wind turbine blade piece Hokuyo LiDAR intensity maps

wind turbine blade surface, a series of test were performed. An example output from the scanning of three different blade segments with varying levels of roughness can be seen in Figure 4.9, together with the blades themselves 4.8. It was determined that that the method can be used for initial scanning and determining if the roughness areas are large enough to warrant further,

more in-depth SfM scanning.

## References

- [1] M. S. Nielsen, I. Nikolov, E. K. Kruse, J. Garnæs, and C. B. Madsen, "High-resolution structure-from-motion for quantitative measurement of leading-edge roughness," *Energies*, vol. 13, no. 15, p. 3916, 2020.
- [2] I. Nikolov and C. Madsen, "Quantifying wind turbine blade surfaces using sandpaper grit sizes," submitted.
- [3] —, "Benchmarking close-range structure from motion 3d reconstruction software under varying capturing conditions," in *Euro-Mediterranean Conference*. Springer, 2016, pp. 15–26.
- [4] Agisoft, "Metashape (photoscan)," <http://www.agisoft.com/>, 2010, accessed: 2020-06-15.
- [5] Bentley, "Contextcapture," <https://www.bentley.com/en/products/brands/contextcapture>, 2016, accessed: 2020-06-15.
- [6] "Hirox rh-2000 microscope," <http://www.hirox-europe.com/products/microscope/RH-2000-digital-microscope>, accessed: 2019-03-06.
- [7] I. Nikolov and C. B. Madsen, "Calculating absolute scale and scale uncertainty for sfm using distance sensor measurements: A lightweight and flexible approach," in *Recent Advances in 3D Imaging, Modeling, and Reconstruction*. IGI Global, 2020, pp. 168–192.
- [8] Hokuyo, "Utm-30lx," <https://www.hokuyo-aut.jp>, 2010, accessed: 2020-03-06.
- [9] Y. Fukuda, K. Kobayashi, K. Watanabe, and T. Kinoshita, "Target object classification based on a fusion of lidar range and intensity data," in *2014 Proceedings of the SICE Annual Conference (SICE)*. IEEE, 2014, pp. 1312–1317.
- [10] S. Khan, D. Wollherr, and M. Buss, "Modeling laser intensities for simultaneous localization and mapping," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 692–699, 2016.

# Chapter 5

## Conclusion

The work presented in this thesis is part of a larger research project connected to wind turbine blade edge surface roughness detection using semi-autonomous drones. The surface roughness is extracted using Structure from Motion 3D reconstruction. To achieve the requirements of the main project five hypotheses were set as starting points. The thesis uses the context of wind turbine blades, as well as a more general research context to prove these hypotheses:

1. It is possible to experimentally map the relationship space between the capturing conditions and setup parameters and the quality of the resultant SfM reconstructions.
2. Capturing sub-millimeter surface details from wind turbine blade surfaces is possible, provided the used hardware can ensure enough information is gathered and the capturing conditions are met.
3. It is possible to automatically scale SfM reconstructed wind turbine blade surfaces to absolute scale, using late data fusion of the reconstructions with positioning or distance data, even if it is not possible to manually measure the real surface.
4. Surface roughness information and reconstruction noise are separable, using 3D mesh and capturing setup analysis.
5. It is possible to quantify and use the reconstructed 3D surface to explain the properties of the real world one.

These hypotheses were proven, as part of the various stages of the SfM capturing, production and analysis pipeline, present in the structure of this thesis, as themes and sub-themes:

Thesis Theme	Hypothesis
<b>Structure from Motion Data Capturing</b>	
Image Acquisition	1, 2
Data Fusion	3
<b>Structure from Motion Data Analysis</b>	
Reconstruction Quality Analysis	4, 2
Reconstructed Surface Classification	5

Hypotheses 1 and 2 provided the base assumptions in the research of SfM. To verify them, the initial steps were to create a number of comprehensive benchmarks and to compare the quality of the output of a number of state of the art SfM solutions under varying capturing conditions - camera positions, overlap, illumination, the use of a turntable, different camera resolutions. Two datasets were created based on the research, containing both image data and ground truth for a straightforward comparison. The real life image acquisition was proven to be time-consuming, so an interactive environment for capturing synthetic images was also developed. It was shown that these images could produce useful reconstruction results, even without using a real-life illumination model, but just an approximation, in a fraction of the time it would take traditional 3D modeling applications. The different camera positions used in the benchmark proved that the quality of SfM output can vary widely, depending on the capturing positions. This prompted the development of a lightweight localization and mapping algorithms that could be used for capturing images from wind turbine blades, using prior information about the shape of the blade. Finally by combining the knowledge from wind turbine blade localization and the requirements for a good SfM reconstruction, a study was conducted on how these positions and the distance from the blade would influence the quality of the SfM output. For this both a 2D shape and 3D surface feature roughness analysis were conducted, showing that distance from the imaged surface is the most important factor, with the amount of imaging positions having smaller impact, but helping with the robustness of the captured data.

To be able to extract correct surface information from the reconstructions, hypothesis 3 needed to be verified. It was seen that capturing the correct absolute real-life scale, was not a trivial problem, especially when there was no easy access to the imaged surface. This prompted a deeper dive into using different types of sensors to provide additional information about the capturing environment and surface. This information can be used to create a late data fusion, which would help with the proper scale of the reconstruction. Two types of sensors were researched - positioning sensors, like GPS and GPS-RTK and distance sensors like ultrasound and LiDAR sensors. It was demonstrated that position sensors could provide an easy way to fuse data

present on every UAV, with SfM data and calculate the absolute scale. The problem with this approach was the large uncertainty of the GPS data and the fact that the positioning measurements could become wrong, if the captured object was moving. These problems were addressed, by using distance sensors, as they provided more precise measurements and could follow the position of the imaged surface. It was also demonstrated that even cheaper distance sensors could be used for this, providing accurate, but less precise results, while more expansive sensors would provide more robust results under less optimal conditions. As part of this research it was also tested how the uncertainty of the calculated scale, would change depending on the uncertainty of the sensor used to capture the additional data. The dataset used for this tests was also published to facilitate experimentation.

Once the correctly scaled 3D surfaces were present, the next steps required further testing of hypothesis 2, as well as the verification of hypothesis 4, for further processing of the data and extracting information about the scanned objects. An initial data analysis was made, in order to separate the surface roughness, from the possible noise and geometric errors that SfM can introduce under sub-optimal conditions. This data analysis used traditional 3D mesh metrics, in combination with metrics specific for the SfM capturing pipeline, like camera positions, overlap, focus, amount of object surface seen from each camera, etc. to segment the reconstruction's surface into noisy and not noisy points. It was demonstrated that the proposed solution can achieve a high degree of accuracy and is general enough that it can be used with objects with different size, shape, color and surface details. Once the usable data was separated from the noise, the quality of that data needed to be evaluated. This was done to test if the data could be used for sub-millimeter surface inspections. To do this a comparison was made between the SfM reconstructions and the same surfaces captured by confocal microscopy. It was shown that by having a camera with high enough resolution, SfM can produce a surface with a level of detail close to that of a microscopy capture - capturing the same roughness structures.

Finally, to demonstrate the validity of hypothesis 5, two different use cases were presented for the surface roughness data coming from SfM. The first one was for classification of the surface roughness of wind turbine blades, using standardized measuring structures in the form of sandpaper with varying grit sizes. For this a number of surface features were extracted from the reconstructed sandpaper patches and used to train a classifier method. The method could then assign a number to each point in the blade surface, depending on how close it resembled that sandpaper grit size. A second use case was presented in the form of using the extracted roughness data for modulating the amplitude and frequency of a vibro-tactile system, to achieve the sensation of touching the reconstructed surface. For this the roughness was classified in different degrees of coarseness. It was demonstrated that an

user could detect different surface roughness values, using only the feedback coming from the vibrations of the system.

The work present in this thesis was based on research work encompassing different topics, fields and use cases. A number of interesting and useful for the scientific community results were achieved in each of the presented topics. Many of the topics have been presented as initial research and are open to follow up work.

A number of directions for future work can be pursued, to build on the already established research presented in this thesis. Three possible further research topics are presented as follows:

- Using deep neural networks and wind turbine blade surface data for detection and classification of surface damage. Building upon the research presented in paper J [1] and paper G [2], which relied on hand crafted features and traditional classification algorithms, a large quantity of wind turbine surface point cloud data can be used instead. Deep learning can be used for detecting and classifying surface damage depending on their size, depth, position on the blade surface.
- The combining of multi-sensory data with SfM reconstructions can be extended. Both 2D and 3D LiDAR data can be used for creating initial lower resolution blade surfaces, while later high resolution data being mapped to the same surfaces at specified points of interest. This way the position of detected roughness can more easily be mapped to the global blade geometry and additional image data can be projected onto the lower resolution parts.
- It was seen that light intensity and direction play integral role in correct roughness detection of smooth surfaces like wind turbine blades. This idea can be extended by exploring shape from shading. [3] paradigms for estimating roughness, by combining camera views with light coming from different directions. These light direction can be both experimentally tested, as well as simulated using ray-tracing and ray-marching algorithms on the 3D reconstructed models.

## References

- [1] I. Nikolov and C. Madsen, "Quantifying wind turbine blade surfaces using sandpaper grit sizes," submitted.
- [2] —, "Rough or noisy? metrics for noise detection in sfm reconstructions," accepted at MDPI Sensors Journal.

## References

- [3] Y. Quéau, J. Mérou, F. Castan, D. Cremers, and J.-D. Durou, “A variational approach to shape-from-shading under natural illumination,” in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*. Springer, 2017, pp. 342–357.

## References

## **Part II**

# **Structure from Motion Data Capturing**



# Paper A

## Benchmarking Close-Range Structure from Motion 3D Reconstruction Software under Varying Capturing Conditions

Ivan Nikolov and Claus Madsen

The paper has been published in the  
*Proceedings of 6th International Euro-Mediterranean Conference* pp.15-26, 2016.

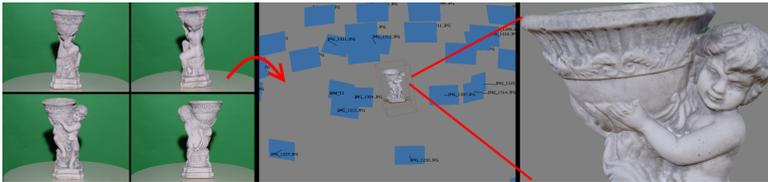
© 2016 Springer  
*The layout has been revised.*

## Abstract

*Structure from Motion 3D reconstruction has become widely used in recent years in a number of fields such as industrial surface inspection, archeology, cultural heritage preservation and geomapping. A number of software solutions have been released using variations of this technique. In this paper we analyse the state of the art of these software applications, by comparing the resultant 3D meshes qualitatively and quantitatively. We propose a number of testing scenarios using different lighting conditions, camera positions and image acquisition methods for the best in-depth analysis and discuss the results, the overall performance and the problems present in each software. We employ distance and roughness metrics for evaluating the final reconstruction results.*

## A.1 Introduction

Structure from Motion (SfM) for 3D reconstruction has come a long way in recent years. The technology is at a point where a multitude of commercial and free packages exist, enabling non-experts to quickly and easily capture high quality models from uncalibrated images. An example is given in Figure A.1.



**Fig. A.1:** Example of 4 out of N input images, taken from different view points and the resulting camera position triangulation and dense point cloud creation. The view is from Agisoft PhotoScan.

Most of these packages are used for landscape reconstruction, creation of orthomosaics and large scale reconstructions. They can be also used for close-range reconstructions. This makes them perfect for use in cultural heritage preservation, artifact digitalization, virtual museums and others. However, many of these solutions come with high initial and upkeep monetary costs. This makes choosing the one most suitable for a specific task an important first step for each project relying on 3D reconstruction - both in result accuracy, resource requirements and performance across varying conditions. Such an endeavour can require a large investment of time. This is why in our paper we provide an in-depth overview of the newest and most widely used commercial software solution tested across various conditions. We concentrate on close-range SfM, as opposed to aerial or long-range.

Six commercial 3D reconstruction software solutions are chosen for testing in the paper. Each of the solutions takes an unordered list of images as input, extracts features and creates a sparse point cloud, triangulating the camera positions. A dense point cloud and a mesh are created by interpolating the sparse point cloud. Texture of the reconstructed object is also created.

Six different objects are used for the reconstructions, depicted in Figure A.2. They are selected according to their varying reconstruction difficulty and different problems that they present like textureless surfaces, repeatable patterns, symmetrical objects, glossiness, etc. Objects are scanned with a white light scanner for evaluating the meshes produced by the SfM packages.

Six different scenarios are tested. These scenarios cover different lighting, positioning and shooting setups. These experiments show that the environmental conditions have a noticeable impact on the final reconstruction and affect some software solutions more than others.

For verifying the accuracy of the output meshes from the different programs, two qualitative methods are chosen: 1) calculating the signed distance between ground truth objects and the reconstructions; 2) comparing the local roughness profiles between the ground truth objects and the reconstructions. The results show that some of the tested packages have more problems reconstructing glossy, symmetrical and textureless surfaces, than others, resulting in complete failures. Some programs sacrifice details for a less noisy final mesh, while others capture more detail, but are very sensitive to noise. A moving camera setup with uniform lighting also gives higher reconstruction accuracy than a turntable setup.

## A.2 Related Work

SfM is just one of many techniques for 3D reconstruction of objects and artifacts. Other techniques are beyond the scope of the paper, but for a quick overview the work in stereo-vision reconstruction [1], structured light [2] or laser scanning [3] is available for reference.

For SfM reconstruction most resources for benchmarks and comparisons are either from archaeological context [4] or from geomapping context [5]. These give valuable information, but are mostly focused on one type of surfaces and objects to reconstruct under a more limited set of environment conditions. Other resources [6] [7] give more in-depth comparison using both their own datasets and freely available ones, but lack the comparison for a larger number of software solutions.

## A.3 Tested Software

We have chosen six of the state of the art software packages for 3D reconstruction. These products are *Agisoft PhotoScan Pro* [8], *Bentley ContextCapture* [9], *Autodesk Memento (ReMake)* [10], *Pix4D* [11], *3Dflow 3DF Zephyr Pro* [12] and *Reality Capture* [13]. For more information on some of the important features each of the selected software solutions has, please refer to Table A.1. The prices are subject to change and are given as they are in the time of writing this paper and converted to dollars. In the output column four of the most widely used ones for close-range photogrammetry are given to preserve space - 3D mesh, texture, sparse point cloud, dense point cloud. Additional outputs like orthophotos, orthomosaic, fly-through videos, depth and normal maps, etc. are supported by many of the programs, but are out of the scope of this paper.

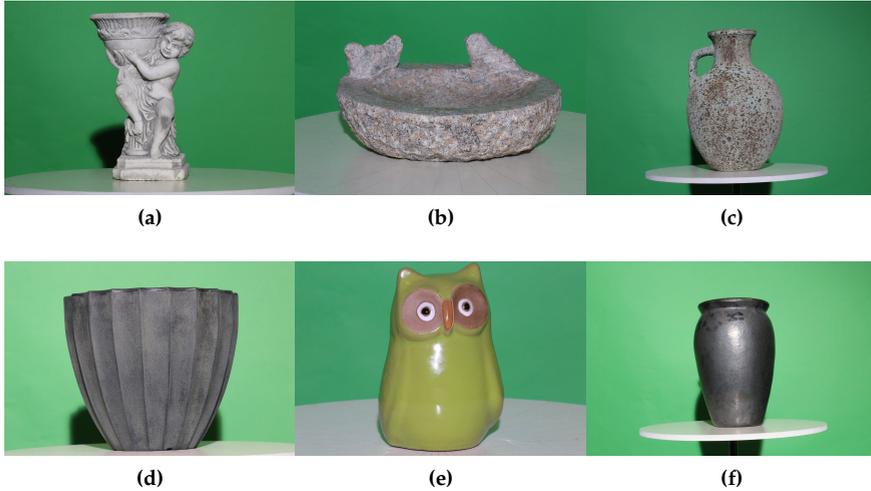
**Table A.1:** Tested software solutions with some of their most important characteristics. In the output column the shortened names denote: dense point cloud (DPC), sparse point cloud (SPC). The price is given for both standard and pro versions. Bolded font denotes the one used for testing.

Program	Outputs	Online/Offline	OS	Scripting
ContextCapture	Mesh/Texture/DPC	Offline	Win	Yes
Memento	Mesh/Texture	Online/Offline	Win/Mac	No
PhotoScan	Mesh/Texture/SPC/DPC	Offline	Win/Mac/Linux	Yes
RealityCapture	Mesh/Texture/SPC/DPC	Offline	Win	Yes
3DF Zephyr	Mesh/Texture/SPC/DPC	Offline	Win	No
Pix4D	Mesh/Texture/DPC	Offline	Win/Mac	No

## A.4 Datasets

The six chosen objects are shown in Figure A.2. These objects are selected based on a number of criteria concerning the properties of the materials that they are made of. These criteria are used to judge the capability of each software to handle different difficult cases, which are considered weak points for SfM. The criteria are as follows - *glossy/smooth surfaces, monochrome colors, very dark/black color, repeating patterns, partial occlusions, symmetrical surfaces*. They may result in failures in reconstruction, decreased overall accuracy, cause holes and noise in the resultant point clouds and mesh [14] [15].

As an initial observation the objects are divided into two groups depending on their perceived reconstruction difficulty. The easy to reconstruct objects - angel statue, sea vase and bird bath and the hard to reconstruct objects - black vase, plastic vase and owl statue. The angel statue and sea vase are perceived as easy because they have a lot of surface detail and features, both global and local, which should make them easy to reconstruct by all the pro-



**Fig. A.2:** Testing Objects: a) angel statue, b) bird bath, c) sea vase, d) plastic vase, e) owl statue, f) black vase. Typical size of the objects is between 25 and 60 cm

grams. They also have some partial self occlusion, which will be tested. The bird bath is also feature rich and has both very smooth and glossy surface parts, as well as rough ones. The black and plastic vases are perceived as hard, because of their color, glossiness and repeated patterns. The owl statue is chosen as an intermediate object, which has a lot of glossiness and feature poor parts, as well as non-glossy more feature-rich ones.

The input images are taken using a *Canon 6D* camera at maximum resolution of  $5472 \times 3648$ . A zoom lens with focal length of  $70 - 300\text{mm}$  is used to accommodate the different zoom levels needed for the different objects. The reconstructions are carried out on a stand alone laptop equipped with *Intel Core i7 - 4710HQ* at  $2.50\text{ GHz}$ ,  $16\text{ Gb}$  of RAM and a *GeForce GTX 970M*. The operating system is Windows 8.1. Each of the six objects has been scanned with a high resolution, high accuracy white light scanner from *Aicon*. These scans are considered detailed enough to be used as a benchmark for the performance. To demonstrate the accuracy and detail of the scans, a cube with known dimensions is also scanned and the measurement of the 3D model's sides are compared to the real world ones. The two differ by an average of  $1.03\text{mm}/1.12\text{mm}/0.93\text{mm}$  in width/height/depth. Henceforth these scans will be referred to as ground truth objects, while the outputs from each of the tested programs will be referred to as reconstructed objects.

## A.5 Testing Scenarios and Results

### A.5.1 Main Test Scenario

All six objects are used in the initial test scenario, together with all the tested programs. The distance between the reconstructed and ground truth objects is calculated, together with the local roughness. The scenario aims to determine how are the selected programs fairing when tested with both easy and hard to reconstruct objects, as well as noting their speed, accuracy and robustness against noise. The test also aims to determine the object factors which make reconstruction hardest for each of the programs.

The test scenario uses photos captured in an indoor controlled environment. The image capturing algorithm is as follows - the captured object is positioned on a turntable; the camera on a tripod is facing the object and is lower than it for capturing the first set of images at lower angle; one light is positioned on a stand above the camera so it shines directly at object; a photo is taken and the turntable is turned  $20^\circ$ ; this is done 18 times, so the object is captured from all sides; the camera on the tripod is then repositioned higher two times, each time 18 more photos are taken; a total of 54 photos of the three different height sets. The *CanonD6* camera is used for taking photos as it gives high detail photos, without straining the hardware of the testing machine.

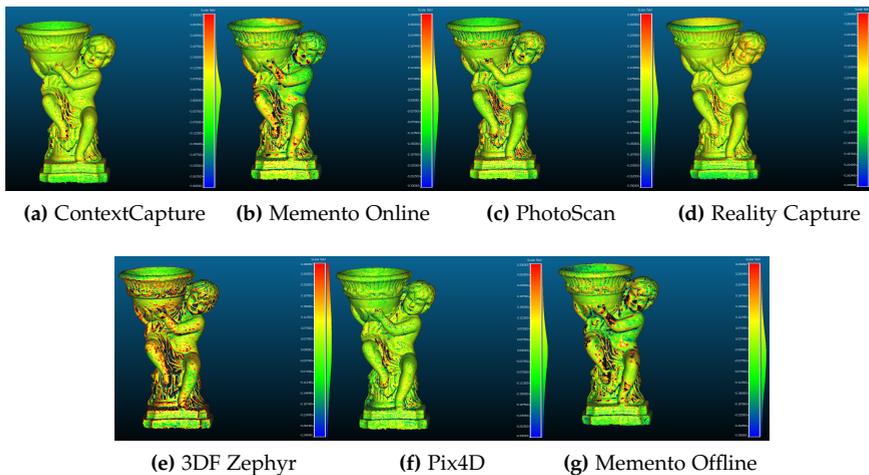
The total processing time of creating the 3D model is noted for each program. For the online version of Memento, the processing time does not give a proper estimate of the working time. It is given just for a more full presentation of the data. This data is given in Table A.2. A coarse visual inspection is done on the created model, focusing on severe problems with the objects.

**Table A.2:** Processing time in seconds for each of the six objects by the tested software solutions. Models which contain problems like missing sides, broken parts, floating noise, etc. are marked with red. Models which could not be reconstructed are given an *N/A* notation.

Program	Angel	Bird Bath	Owl	Sea Vase	Plastic Vase	Black Vase
ContextCapture	2820 sec	3600 sec	N/A	3780 sec	3060 sec	N/A
Memento Online	4860 sec	4920 sec	5160 sec	4440 sec	4260 sec	5340 sec
PhotoScan	4020 sec	4500 sec	3780 sec	4560 sec	4740 sec	3480 sec
RealityCapture	5220 sec	6480 sec	N/A	6720 sec	2820 sec	N/A
3DF Zephyr	3720 sec	4440 sec	4140 sec	4860 sec	3060 sec	4680 sec
Pix4D	4140 sec	3240 sec	N/A	4860 sec	3960 sec	3720 sec
Memento Offline	11520 sec	9360 sec	7140 sec	10320 sec	7980 sec	N/A

All packages, except Memento offline have comparable processing times, which depend on the complexity of the reconstructed object. Memento online, PhotoScan and 3DF Zephyr could reconstruct all objects, while ContextCapture, Reality Capture and Pix4D experienced the most problems. The coarse visual inspection is followed by a more qualitative inspection, using

the ground truth scanner data for comparison. The idea suggested by Schönig and Heidemann [6] is used for this part of the test scenario. In their paper they conclude that each tested program produces 3D models and point clouds of different density, which also may contain parts of the background or noise particles. Therefore, it is better to use the reconstructed models as reference and compare the ground truth data to each, noting the difference. In addition their idea of using the meshes for comparison is used, as opposed to using the point cloud. This gives the possibility to test signed distances using the model’s pre-calculated normals.



**Fig. A.3:** Pseudo color distance maps between the ground truth and the reconstructed objects. Red colors indicate distances above the ground truth, blue colors indicate distances below the ground truth and green colors indicate where the surfaces coincide.

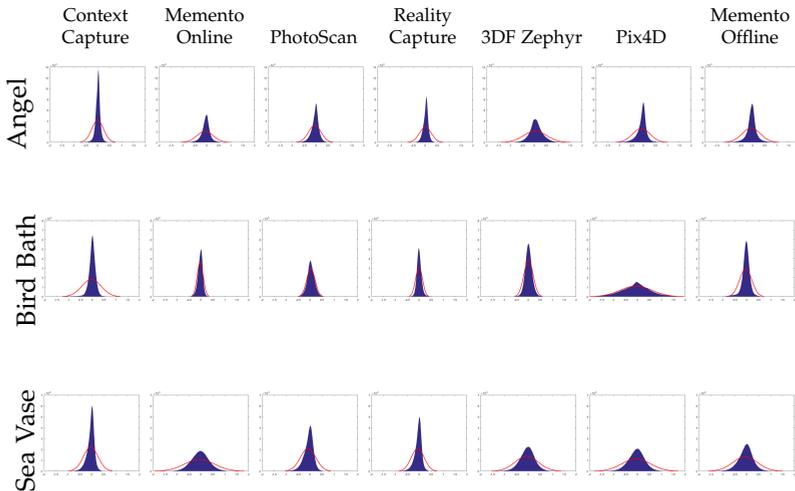
The comparison between the reconstruction and the scanned data is done using the free open source software *CloudCompare* [16]. The reconstructed models are scaled to the absolute scale of the ground truth and registered to it using an iterative closest point algorithm (ICP) by Besl and McKay [17]. Once the models are registered the distances between the triangles of the reconstructions and the ground truth is calculated. Using the normals of the meshes the distance is calculated as signed. These distances are visualized as pseudo color heat map. The pseudo color maps for the angel statue can be seen in Figure A.3. The maps are filtered removing distances outside the interval of  $[-0.3mm; 0.3mm]$ , for easier visualization. From the distances, the mean and standard deviation of the distance distribution for the whole object are calculated. A Gaussian normal distribution is assumed for the modelling of the distance distribution between the ground truth and the reconstruction. The mean and standard deviation are given in Table A.3 for the easy to re-

## A.5. Testing Scenarios and Results

construct objects - angel statue, bird bath and sea vase, together with the Gaussian distributions for them in Figure A.4. For the hard to reconstruct objects - the plastic vase, owl and black vase the data is given in Table A.4 and Figure A.5, respectively.

**Table A.3:** Mean value ( $\mu$ ) in  $mm$  and standard deviation ( $\sigma$ ) in  $mm^2$  of the distance metric for each software solution for the three objects selected as easy to reconstruct

	Angel	Bird Bath	Sea Vase
	Mean/Variance	Mean/Variance	Mean/Variance
ContextCapture	-0.024/0.703	-0.030/0.588	-0.245/2.016
Memento Online	-0.089/0.438	-0.039/0.382	-0.408/2.277
PhotoScan	-0.109/0.805	0.034/0.175	-0.463/2.321
RealityCapture	-0.038/0.486	-0.006/0.143	-0.481/2.421
3DF Zephyr	-0.040/1.020	-0.045/1.537	-0.911/3.514
Pix4D	-0.194/1.124	-0.060/0.668	-0.425/2.419
Memento Offline	-0.080/0.569	-0.046/0.40	-0.255/2.983

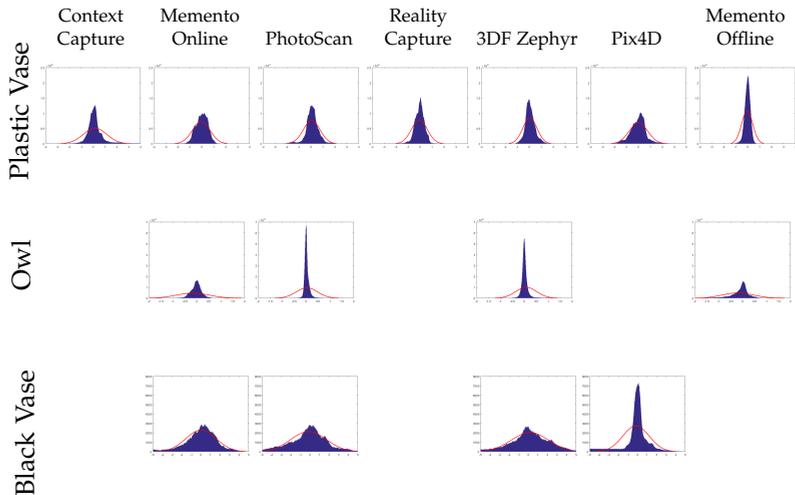


**Fig. A.4:** Histograms of the Gaussian distribution characterizing the distances between the ground truths and the three easy objects. All the histograms are scaled the same.

The initial speculation dividing the objects into easy and hard ones is proven by the amount of reconstruction failures. Both the black vase and the owl statue, experience much higher number of failures, compared to the other objects. The plastic vase fairs better, but because of its symmetrical featureless and dark surface, the reconstruction suffers from improperly placed geometry. This can also be seen from the Gaussian histogram distributions in Figure A.5, where the distributions for both the black vase and the plastic vase are much wider, showing larger deviation distances from the ground

**Table A.4:** Mean value ( $\mu$ ) in  $mm$  and standard deviation ( $\sigma$ ) in  $mm^2$  of the distance metric for each software solution for the three objects selected as hard to reconstruct

	Plastic Vase	Owl	Black Vase
	Mean/Variance	Mean/Variance	Mean/Variance
ContextCapture	-2.512/10.601	N/A	N/A
Memento Online	-3.450/6.697	-0.937/3.318	-4.549/5.886
PhotoScan	-3.791/7.027	0.371/6.806	-4.331/5.758
RealityCapture	-4.395/7.222	N/A	N/A
3DF Zephyr	-4.814/7.471	0.169/3.191	-4.035/5.933
Pix4D	-3.782/7.187	N/A	-4.794/6.027
Memento Offline	-5.074/7.429	-0.929/0.977	N/A

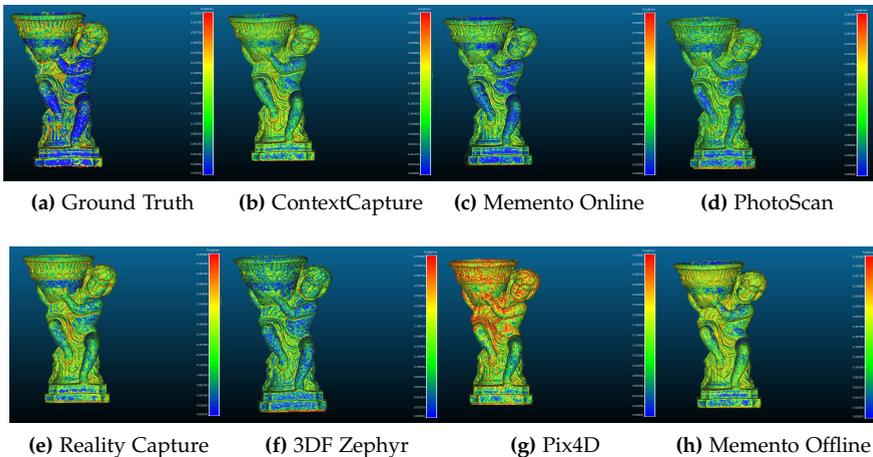
**Fig. A.5:** Histograms of the Gaussian distribution characterizing the distances between the ground truths and the three hard objects. The missing histograms are programs which failed to reconstruct the object. All the histograms are scaled the same.

truth. The owl statue has less noisy histogram, but it suffers from holes in the reconstruction. ContextCapture and Reality Capture demonstrate the overall smallest mean and variance deviations from the ground truth for the easy objects, but both programs completely or partially fail when the surfaces are not optimal. 3DF Zephyr, Memento Online and PhotoScan on the other hand are much more consistent and have a more graceful degradation of performance, but tend to miss smaller details and have an overall high variance in the distance distribution. From here another observation can be made - the programs can be roughly divided into ones that capture a lot of small detail at the price of noise and easier failures - Context Capture, Reality Capture, Memento Offline and the ones that are more consistent and robust, but fail to capture details - Memento Online, 3DF Zephyr, PhotoScan. Pix4D is mainly

## A.5. Testing Scenarios and Results

aimed at aerial photos and this clearly shows, as the program is much noisier in all instances.

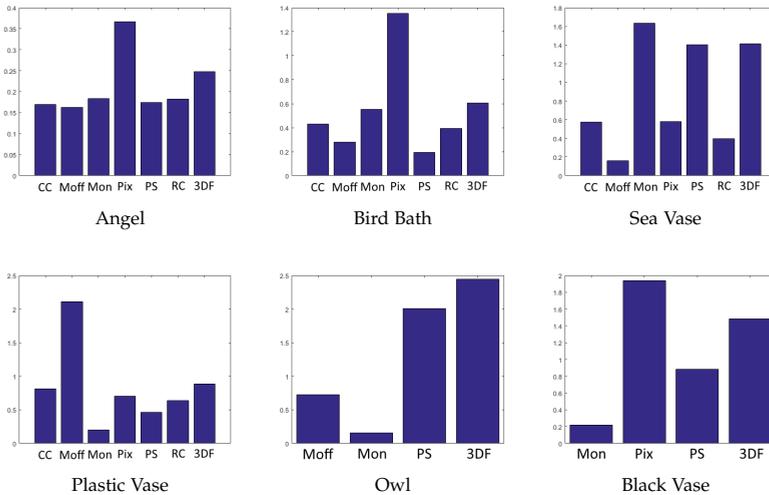
To determine the amount of noise and over-smoothing of features in the reconstructions compared to the ground truth, a second metric is introduced. The local roughness of both the reconstructions and the ground truth is calculated using the Gaussian curvature of the models, which is normalized to give proper weights to rough patches and smooth patches near edges. The method is introduced by Wang et al. [18] for assessment of mesh visual quality. The method is useful in the case of our paper as it generates an accurate roughness map, which can give both visual and more quantitative information for the success of the reconstruction. The roughness map is also visualized as a pseudo color map, which is given in Figure A.6.



**Fig. A.6:** Pseudo color roughness maps of the ground truth and reconstructed meshes. The colors go from red to blue through green, depending on how rough the surface is.

From the local roughness map, the histograms of both the ground truth and the reconstructions is calculated. Using these histograms the Kullback-Leibler distances [19] between the ground truth and reconstructions are calculated. This gives a measurement of the similarity between the two, which penalizes deviations from the roughness of the ground truth both caused by introduction of noise in the reconstructed mesh and in over-smoothing details in it. Figure A.7 has the results from the roughness histogram distances, where smaller values give more faithful reconstructions, roughness-wise.

The results from the roughness metric support the division of the programs. Pix4D introduces a lot of noise and smooths details. This can be seen in both Figure A.6 and Figure A.7, where it has clear disadvantage in many of the cases. Figure A.6 also shows that Pix4D, Memento Offline and



**Fig. A.7:** Bar chart visualizing the calculated Kullback-Leibler distances between the roughness histogram of the ground truth and the reconstructed objects. The tested software is denoted with short names - *Context Capture (CC)*, *Memento Online (Mon)*, *PhotoScan(PS)*, *Reality Capture (RC)*, *3DF Zephyr (3df)*, *Pix4d (Pix)* and *Memento Offline (Moff)*.

Reality Capture have introduced a lot of noise on the smoother parts of the angel, compared to the ground truth, like the stomach and legs, while Memento Online, 3DF Zephyr and PhotoScan have smoothed out small features in the face and hair of the angel. Memento Offline, Reality Capture and PhotoScan manage to capture most of the detail on the easier to reconstruct objects like the angel and the bird bath, without introducing too much noise as evidenced by the smaller histogram distances. However they fail on the smoother objects like the plastic vase and the owl, where they introduce uncertainty noise. Memento Online and 3DF Zephyr tend to over-smooth the surfaces as evidenced by the bar chart of the sea vase.

## A.5.2 Follow Up Test Scenarios

One of the best performing objects - the bird bath is tested in a number of follow up scenarios under different capturing conditions. This is done to determine the effect of capturing conditions on the reconstruction results. Five follow up experiments are carried out focusing on different combinations of conditions. First tested condition is the effect of rotating the camera to capture images from different views, as opposed to using a turntable to rotate the object and keep the camera stationary. This test aims to assess if a moving background and completely static lighting will help with reconstruction process, as opposed to the lighting which "moved" with the object in the case of

using a turntable. The second tested condition is using multiple light sources for a more even lighting, as opposed to one directional light. The third condition is using different number of photo positions, combined into bands of photos, with varying height. Five and three bands of photos are created. The first two contain 18 photos each taken in  $20^\circ$  intervals, the next two contain 9 photos in  $40^\circ$  and the final one contain 4 images. The setup also aims to test if introducing information from more angles can help the feature point matching algorithm of the tested software solutions. The same analysis pipeline is used as with the main experiment, using the ground truth scans to compare with. The results from the different combination of conditions are given in Table A.5.

**Table A.5:** Mean value ( $\mu$ ) in  $mm$  and standard deviation ( $\sigma$ ) in  $mm^2$  of the distance metric for the bird bath object for each of the tested software solution from the five tested shooting scenarios. The tested software is denoted with short names - *Context Capture (CC)*, *Memento Online (Mon)*, *PhotoScan(PS)*, *Reality Capture (RC)*, *3DF Zephyr (3df)*, *Pix4d (Pix)* and *Memento Offline (Moff)*.

	No Turntable		Turntable		
	Multiple Lights		One Light	Multiple Lights	
	Five Bands	Three Bands	Five Bands	Five Bands	Three Bands
	$(\mu)/(\sigma)$	$(\mu)/(\sigma)$	$(\mu)/(\sigma)$	$(\mu)/(\sigma)$	$(\mu)/(\sigma)$
CC	-7.167/13.289	-4.147/8.052	N/A	N/A	N/A
Mon	-0.209/2.028	-0.094/1.306	-0.366/2.148	-0.947/3.826	-0.309/2.305
PS	-0.283/2.312	-0.240/2.685	0.206/1.982	-0.212/2.410	-0.167/1.159
RC	-0.031/0.284	-0.014/0.689	N/A	0.108/1.710	N/A
3df	-0.039/0.584	0.011/0.411	-0.308/2.035	-0.372/2.712	-0.165/0.922
Pix	-0.169/0.407	-0.166/1.911	-5.023/12.401	-0.204/1.520	0.061/1.674
Moff	-0.105/1.355	-0.071/1.118	-1.135/4.253	-0.389/2.262	-0.114/1.552

The tests show that using static lighting and moving background without a turntable yields a higher accuracy, with lower mean and standard deviation values, compared to the turntable results. There is also a difference between using multiple light sources and just one directional one, with the latter introducing more noise, which can be seen by the higher standard deviation in the table above. This shows that if higher accuracy is necessary, a capturing process without a turntable and with uniform lighting and more diverse camera positions need to be used, even if this will cost more time and resources.

## A.6 Conclusion and Future Work

Our paper presents a head to head comparison of the state of the art SfM 3D reconstruction software solutions. As part of the research we tested six programs - ContextCapture, PhotoScan, Memento, Reality Capture, Pix4D and 3DF Zephyr. We tests the programs on both a variety of challenging objects and on images taken from different capturing conditions. Reconstruction re-

sults were evaluated against ground truth objects on the basis of distance measurement and roughness comparison.

We demonstrated that programs can be roughly divided in two groups - ones that are more robust to sub-optimal objects and capturing conditions, but do not manage to capture smaller details and ones that can capture high amount of details, but degrade in performance and introduce a lot of noise, once the optimal conditions are not met. Additionally we show that using a turntable can have a negative effect on the accuracy of the reconstructed objects, as well as using a single light source. For optimal capture conditions a moving camera, multiple lights and images taken from multiple locations and angles are recommended.

As an extension to this paper we propose introducing prior information to the programs like - camera positions, feature points, markers, etc., as well as combining multiple software solutions in a pipeline for achieving better results and helping failed reconstruction attempts on hard to reconstruct objects. Testing different cameras is also proposed as an extension to the different environment conditions.

## References

- [1] A. H. Ahmadabadian, S. Robson, J. Boehm, M. Shortis, K. Wenzel, and D. Fritsch, "A comparison of dense matching algorithms for scaled surface reconstruction using stereo camera rigs," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 78, pp. 157–167, 2013.
- [2] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison *et al.*, "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 2011, pp. 559–568.
- [3] H. Kaartinen, J. Hyypä, A. Kukko, A. Jaakkola, and H. Hyypä, "Benchmarking the performance of mobile laser scanning systems using a permanent test field," *Sensors*, vol. 12, no. 9, pp. 12 814–12 835, 2012.
- [4] F. Nex, F. Remondino, M. Gerke, H.-J. Przybilla, M. Baumker, and A. Zurhorst, "Isprs benchmark for multi-platform photogrammetry." *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 2, 2015.
- [5] A. Koutsoudis, B. Vidmar, G. Ioannakis, F. Arnaoutoglou, G. Pavlidis, and C. Chamzas, "Multi-image 3d reconstruction data evaluation," *Journal of Cultural Heritage*, vol. 15, no. 1, pp. 73–79, 2014.

## References

- [6] J. Schönig and G. Heidemann, "Evaluation of multi-view 3d reconstruction software," in *International conference on computer analysis of images and patterns*. Springer, 2015, pp. 450–461.
- [7] M. Smith, J. Carrivick, and D. Quincey, "Structure from motion photogrammetry in physical geography," *Progress in Physical Geography*, vol. 40, no. 2, pp. 247–275, 2016.
- [8] Agisoft, "Metashape (photoscan)," <http://www.agisoft.com/>, 2010, accessed: 2020-06-15.
- [9] Bentley, "Contextcapture," <https://www.bentley.com/en/products/brands/contextcapture>, 2016, accessed: 2020-06-15.
- [10] Autodesk, "Remake(formally known as memento)," <https://memento.autodesk.com/about>, 2015, accessed: 2020-06-15.
- [11] Pix4D, "Pix4dmapper," <https://pix4d.com/>, 2011, accessed: 2020-06-15.
- [12] 3dFlow, "3df zephyr," <http://www.3dflow.net/>, 2011, accessed: 2020-06-15.
- [13] CapturingReality, "Reality capture," <https://www.capturingreality.com/>, 2016, accessed: 2020-06-15.
- [14] G. Guidi, S. Gonizzi, and L. Micoli, "Image pre-processing for optimizing automated photogrammetry performances," in *ISPRS Technical Commission V Symposium*, vol. 2. ISPRS, 2014, pp. 145–152.
- [15] C. Nicolae, E. Nocerino, F. Menna, and F. Remondino, "Photogrammetry applied to problematic artefacts," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, no. 5, p. 451, 2014.
- [16] D. Girardeau-Montaut, "Cloudcompare-open source project," *Open-Source Project*, 2011.
- [17] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–606.
- [18] K. Wang, F. Torkhani, and A. Montanvert, "A fast roughness-based approach to the assessment of 3d mesh visual quality," *Computers & Graphics*, vol. 36, no. 7, pp. 808–818, 2012.
- [19] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

## References

# Paper B

## Interactive Environment for Testing SfM Image Capture Configurations

Ivan Nikolov and Claus Madsen

The paper has been published in the  
*Proceedings of 14th International Joint Conference on Computer Vision, Imaging  
and Computer Graphics Theory and Applications* pp. 317-322, 2019.

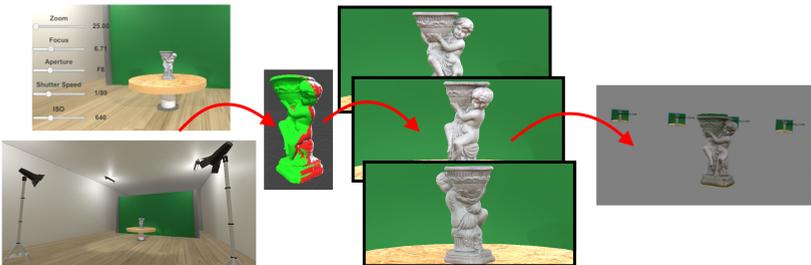
© 2019 SCITEPRESS Digital Library  
*The layout has been revised.*

## Abstract

*In recent years 3D reconstruction has become an important part of the manufacturing industry, product design, digital cultural heritage preservation, etc. Structure from Motion (SfM) is widely adopted, since it does not require specialized hardware and easily scales with the size of the scanned object. However, one of the drawbacks of SfM is the initial time and resource investment required for setting up a proper scanning environment and equipment, such as proper lighting and camera, number of images, the need of green screen, etc, as well as to determine if an object can be scanned successfully. This is why we propose a simple solution for approximating the whole capturing process. This way users can test fast and effortlessly different capturing setups. We introduce a visual indicator on how much of the scanned object is captured with each image in our environment, giving users a better idea of how many images would be needed. We compare the 3D reconstruction created from images from our solution, with ones created from rendered images using Autodesk Maya and V-Ray. We demonstrate that we provide comparable reconstruction accuracy at a fraction of the time.*

## B.1 Introduction

Capturing 3D models of objects has become an important part of the entertainment [1], medical [2] and manufacturing industries [3]. Having not only 2D representations of the objects through images, but a whole 3D model can give more information about the object's appearance, form and scale. When a high level of accuracy is needed in the captured 3D model, the go to technology has been laser scanners [4] and structured light scanners [5], as well as structure from motion [6]. This paper focuses on SfM.



**Fig. B.1:** Overview of our proposed solution. A Unity interactive testing environment, an approximated DSLR camera with changeable settings and a visualization of how much of the object's surface has been scanned. The captured images can then be used for SfM reconstruction testing purposes.

SfM works by first taking images all around the desired object, covering its

whole surface. Features points are extracted from each image and matched between images. By triangulating these matched 2D points on the images, the 3D world coordinates of each camera position, as well as a sparse point cloud of the scanned object can be calculated. The camera positions and the sparse point cloud are then adjusted and interpolated to create a much denser point cloud, which captures a lot of the details of the object. Currently there exist multiple commercial [7], [8] and open-source [9], [10] solutions for SfM reconstruction.

Here comes one of the biggest drawbacks of SfM - the reliance on the quality of the input images. If problems like lack of enough images, blurriness, over/underexposure or noise are present in the input images, they will result in lower quality or complete failure of the reconstruction. Further problems can arise if the captured object has a specular surface, transparent parts or lacks a detailed surface. Testing different configurations of the capturing environment, camera settings, capturing conditions and objects can take a lot of time and can easily become costly if equipment needs to be changed or if the captured object needs to be processed to make its surface more diffuse. Additionally, different SfM solutions have varying degrees of robustness to these problems, making it crucial to know what is the best setup for the task at hand. A lot of research [11], [12] has gone into looking into how all these factors contribute to the output of SfM.

Here comes one of the biggest drawbacks of SfM - the reliance on the quality of the input images. If problems like lack of enough images, blurriness, over/underexposure or noise are present in the input images, they will result in lower quality or complete failure of the reconstruction. Further problems can arise if the captured object has a specular surface, transparent parts or lacks a detailed surface. Testing different configurations of the capturing environment, camera settings, capturing conditions and objects can take a lot of time and can easily become costly if equipment needs to be changed or if the captured object needs to be processed to make its surface more diffuse. Additionally, different SfM solutions have varying degrees of robustness to these problems, making it crucial to know what is the best setup for the task at hand. A lot of research [11], [12] has gone into looking into how all these factors contribute to the output of SfM.

## B.2 Our Proposed Solution

The normal way to test out different capturing conditions and setups is by rendering out images from a 3D model in programs such as Autodesk Maya [13]. This way the user can be in control and change lighting conditions, camera positions, change the environment, etc. This can produce photorealistic results, but has the shortcoming that it requires in-depth knowledge of

the used software solution, there is no easy way to observe the capturing progress. Rendering each image, also takes a long time, making the whole process cumbersome.

This is why we introduce a solution, which aims to address all the above shortcomings and deliver results which are comparable. We propose the creation of a testing environment in Unity, which can approximate the physical properties of a real world capturing setup for taking synthetic images. This environment can be used for initial testing with different setup variations, camera settings, objects, number of images, so a deeper insight can be achieved in the problem, before spending too much time and resources. In addition we propose a intuitive visualization of how much of the scanned object is captured with each image, as well as how much of an overlap there is between the images. An overview image of our proposed solution can be seen in Figure B.1.

We compare our solution's results to both reconstruction results from real life images, as well as to results from synthetic images produced with Maya and V-Ray [14], by using a ground truth created with a structured white light scanner. We demonstrate that our method produces comparable results to the offline rendering approach in a fraction of the time and captures the overall shape and detail present in the real life image reconstruction.

In addition we give some use cases for SfM capturing, where our proposed solution can come in handy and introduce some quality of life functions, which will make the normally tedious and long process easier.

## B.3 Methodology

To create the testing environment each of the parts of a capturing setup needs to be modeled - the camera, the environment and the scanned object. The implementation of each of these is explained in detail in the subsections below. As DSLR cameras are the most widely used type of cameras for SfM 3D reconstruction, the testing environment's cameras are model after the typical DSLR parameters.

### B.3.1 Camera Approximation

Our proposed solution does not aim to simulate how the physics of a real camera work. As mentioned before this has been implemented to a much greater extend in V-Ray for Autodesk Maya and will be extremely challenging to implement in Unity and even more to make it work in real time. This is why we choose to simply model how the different parameters of the camera can change the output image in appearance. The image itself is a "screenshot" of what a Unity camera renders of the environment and the parameters of

the camera change this "screenshot" by introducing standard Unity shader effects, to mimic the real world changes.

A number of camera parameters and functions are modeled for approximating the results from changing them on the final image - aperture, shutter speed and ISO, as well as focal length and depth of field. The design consideration while implementing each are given in the sections below.

### **Focal length and depth of field**

To approximate the change of the field of view and zoom level, when adjusting the focal length, Equation B.1 is used. In the equation  $h$  is the sensor height and  $f$  is the current focal length. The modeled camera's sensor size is given as an input and can be changed depending on the modeled camera and is given in *mm*. The calculated field of view can be clamped to specific values to suit the needs of the testing scenario and to best approximate the effect of using a specific lens with the camera.

$$fov = 2\arctan\left(\frac{h}{2f}\right) \quad (\text{B.1})$$

To mimic more closely an output image from a real camera, the radial barrel and pincushion distortions are also implemented. Both distortions are implemented by modifying the fisheye standard asset shader. As an extension of this, future work is planned to use camera calibration algorithms on a number of cameras and lenses to estimate and better model the intrinsic camera parameters and distortions.

The changing of the depth of field when focusing is done using the depth of field shader that comes with the standard assets. Changing the focus of the approximated camera, changes the calculated shader distance from the camera to the Unity environment, which in turn determines the far plane, beyond which the shader applies a disc shaped blur filter.

### **Aperture, shutter speed, ISO**

Each of the camera settings, change the intensity of the effects that they introduce to the final rendered image. The steps are taken from [15], which are used in most of the state of the art DSLR cameras. The aperture is in the interval between  $[f/1.2; f/64]$ , the shutter speed is in  $[20; 1/8000]$ , while the ISO is in the interval  $[100; 51200]$ .

To properly approximate how each of them affects the final exposure the Additive system of Photographic EXposure (APEX) standard is taken as a starting point [16]. It treats each of the parameters as an additive system, in which the increase or decrease of one, results in doubling or halving the exposure. Equation B.2 shows the relation between the exposure value  $EV$ ,

### B.3. Methodology

the shutter speed value  $T_v$  and aperture value  $A_v$ , and the ISO sensitivity  $S_v$  and brightness value  $B_v$ .

$$E_v = A_v + T_v = B_v + S_v \quad (\text{B.2})$$

The aperture, shutter speed and ISO components are given in Equation B.3. In the aperture equation, the square of the aperture is taken, as the whole area is needed. For the ISO the equation contains  $N$ , which is the constant that gives the relation between the arithmetic sensitivity value and the value used by the APEX standard, while  $S_x$  is the arithmetic sensitivity value. Each of the equations takes the base-2 logarithm to make the equation behave linearly.

$$A_v = \log_2(A^2), T_v = \log_2(1/T), S_v = \log_2(NS_x) \quad (\text{B.3})$$

Finally, the brightness value  $B_v$ , is simplified for the Unity approximation, as it is calculated as a sum of the intensity values of each light source in the Unity scene. This is done to approximate the illuminance of the scene. To approximate the effect of changing exposure, the exposure/brightness shader is used from the standard shader package, with its value calculated from the APEX equation.

In addition to changing the perceived scene exposure, each of the three parameters gives other effects. With lowering the aperture size, the blur disc size is made larger, allowing more of the scene to come into focus and vice versa with increasing the aperture size. In addition, with lowering the aperture value, a blur effect is added to the final render to simulate the possible lens diffraction problem that can arise when the size of the aperture becomes smaller [17].

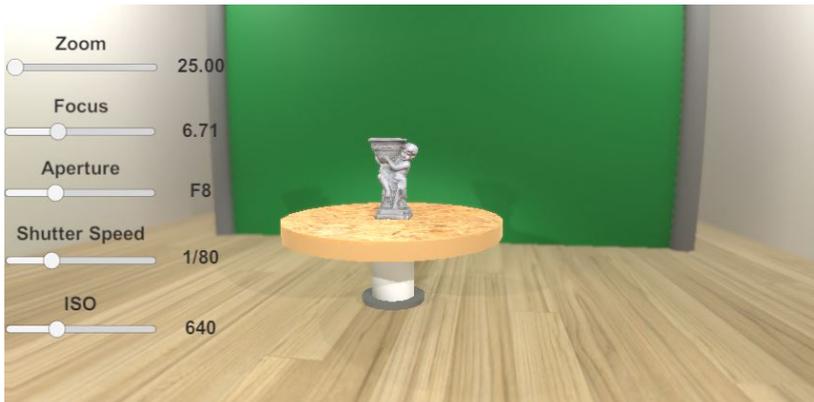
Changing the shutter speed changes the amount of blur present in the final rendered image, if the object or the camera are in motion when the image is taken. This is modeled by introducing the motion blur effect from the standard assets.

Changing the different camera parameters is done by switching to the designated parameter view and moving the specific sliders for each. Figure B.2 shows the camera parameter view.

#### B.3.2 Environment Approximation

The most important parts of a SfM capturing setup for small scale objects are approximated - the lighting, the background and the way to capture different parts of the scanned object. Each of these parts is developed to be fast and easy to use.

The lighting is modeled as both an ambient lighting as well as directed lights. For ambient lighting a number of point lights are used all around the



**Fig. B.2:** The camera parameter view. Each of the main camera parameters, plus the focal length and lens focus can be tweaked from this view, through the use of the sliders.

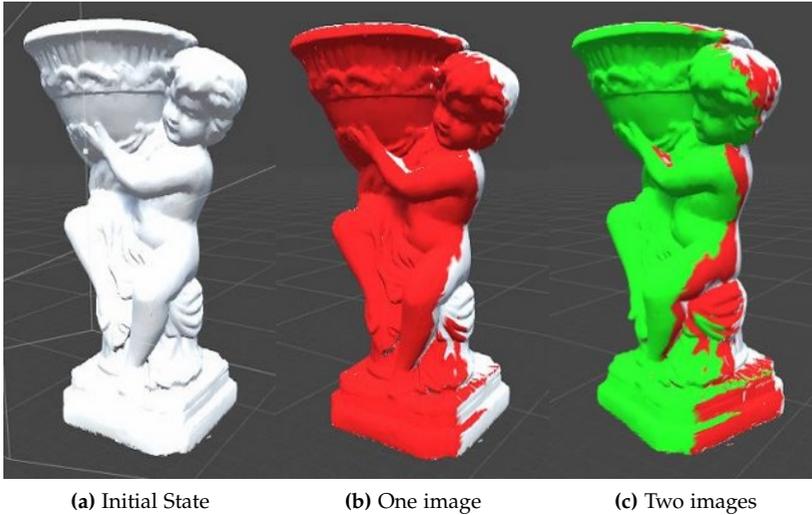
modeled studio. For a harder directed light, a number of directional lights can be setup. The number and position of both types of lights can be changed by the user as needed, as well as the intensity and warmth of the produced light. Soft shadows are rendered for all the objects in the room.

A turntable is implemented in the middle of the capturing room and the object for scanning is placed on it. Finally, a green screen is implemented for use whenever masking is necessary.

### B.3.3 Object Approximation

The user can load the desired object into the environment, which is placed directly on the turntable. As the tested object only exists in the real world, there can be a number of solutions for substitutes used in the environment. A coarsely reconstructed version of the final object or a primitive object such as a sphere, cube, cylinder can be a substitute.

Each time a photograph image is rendered, the seen faces of the captured object from the camera are calculated using an matrix of raycasts from the camera. The object's material can be switched between normal textured view and a view of the seen faces. In the special view, initially the object is plane white. The faces that have been seen from one camera view are colored red, while faces that have been seen from more than one are colored green. Figure B.3 shows the the object view and the coloring as more images are taken with enough overlap.



**Fig. B.3:** The object coverage view. Initially (B.3a) the whole object's surface is white. After the first image, the seen surface is painted red B.3b. After the second image the parts that have been seen from two or more different camera views are colored green B.3c, indicating that there is overlap on the captured images.

## B.4 Solution Test and Results

We choose to compare a SfM reconstruction produced by images from our proposed testing environment against ones rendered using one of the most widely used ways to simulate a physical camera and an image taking setup - Autodesk Maya and V-Ray. In addition a reconstruction is done using real life DSLR camera images as a base case.

For the test real life object a stone angel statue is selected, which can be seen in Figure B.4. The three reconstructed meshes need to be compared to a ground truth model. A high accuracy ground truth is produced using a white light scanner.

The next step is to create a real life image capturing setup. A Canon 6D DSLR camera is used. The camera is a full-frame camera with a sensor size of  $35.8mm \times 23.9mm$ . The photos are taken with the maximum possible resolution of the camera -  $5472 \times 3648$  pixels. The camera is positioned on a tripod in front of a turntable with the captured object. Two Elinchrom D-Lite RX4 lights are setup on both sides behind the camera and are targeted towards the object. A green screen is set behind the object. A photo is taken and the turntable is rotated each time 20 degrees, until the whole object has been captured in 360 degrees, which gives a total of 18 images.

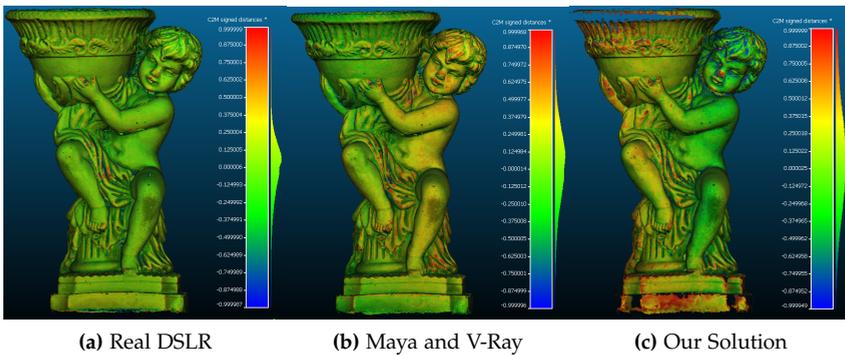
The same capturing setup is created both with our proposed solution and



**Fig. B.4:** Images used for the reconstruction test. Figure B.4a, is the real life image taken from the 6D DSLR camera, B.4b is the rendered image from Maya and V-Ray and B.4c is the image from our proposed solution.

in Maya. In Maya, the physical camera in V-Ray is used for simulating the Canon 6D with the camera parameters saved from the real life capture. The same parameters are used in our environment. The resultant images from the real life setup, the Maya and V-Ray setup and our proposed solution can be seen in Figure B.4.

For each of the three sets of images, the reconstruction is done using Photoscan [8]. The program is chosen as it is frequently used by researchers and provides robust and accurate results compared to other state of the art solutions [11].



**Fig. B.5:** Heat map of the distances between the ground truth object and the reconstruction. Green indicates that the two coincide, while red and blue indicate larger positive and negative distances between the two.

The three reconstructions are compared to the ground truth scan. The open source program CloudCompare [18] is used for the comparison. Each of the reproduced meshes is scaled and registered to the ground truth object. The signed distances between the faces of the reconstructions and the ground truth are calculated. These distances are visualized as a heat map on Figure

## B.5. Use Cases

B.5, where the blue color shows distances which are below the ground truth and the red color - distances above the ground truth, while green indicates that the two surfaces match. From these distances, the mean and standard deviation are calculated for each reconstruction. These are shown in Table B.1.

**Table B.1:** Mean in *mm* and standard deviation in *mm* of the distance metric for the three types of input data - images from a real DSLR Canon 6D camera, rendered images from Maya and V-Ray and our proposed solution

Solution	Mean [ <i>mm</i> ]	Std. Dev. [ <i>mm</i> ]
Real DSLR	-0.38	2.41
Maya + V-Ray	-0.43	2.50
Our Solution	-0.49	3.51

The table shows that the difference between the reconstruction from the real DSLR images and the rendered V-Ray images is negligible, as expected. Only the standard deviation from rendered images is larger, mostly because the texture on the 3D model has lost some of its detail and has had some noise added. This is because the images from our solution lack the fidelity of the other images, as well as the smaller details that would come out from proper rendered lighting calculation. On the other hand, the Maya + V-Ray solution took almost 20 hours to render, making it less than ideal for fast testing and prototyping, compared to the 5 min it took to set the camera settings, find the proper camera position and take the images through our interactive environment.

## B.5 Use Cases

Going from the work of [19], where multiple SfM solutions have been tested under varying environmental and object conditions, it is apparent that a lot of time and work goes into creating a proper setup for a good 3D reconstruction. It is also seen that problems with the camera, the lighting or the capturing environment can drastically lower the quality of the produced model. This is why we introduce a lot of quality of life features in our proposed solution, which aim to make the capturing and testing process easier.

The camera can be made stationary with the turntable rotating a specified number of degrees, until the whole object has been scanned. The second way is by keeping the object stationary, both the standard First Person Shooter (FPS) controller or the flight controller can be attached to the approximated camera, so the user can move manually to the desired positions and rotations.

The implemented green screen can be toggled on and off and its color changed to better contract the color of the object being captured. The lighting

can be moved and the intensity of the light can be regulated both for the directional and point lights, to test different possible illumination setups.

To make it easier for users to judge how much of the object’s surface has been captured from each image an additional visualization mode is implemented. An important part of performing a successful SfM 3D reconstruction, is providing enough images, covering the whole surface of the object and having enough overlap between them.

## B.6 Conclusion and Future Work

In this paper we proposed a interactive testing environment for capturing images for SfM reconstruction. Our solution provides an approximation to the way images of a real life DSLR camera will look like and how the final image changes depending on the camera settings, focal length and focus. Together with the approximated camera we introduce a capturing environment, which can be interactively changed by the user, to accommodate different testing scenarios. Finally we added the possibility for the user to visualize how much of the scanned object’s surface has been captured with each photo and is there overlap between different photos.

We tested our solution’s output against an offline rendering output produced by Autodesk Maya and V-Ray and demonstrated that we achieve similar results at a fraction of the time.

For future work we would like to remake the interactive testing environment in another engine like Unreal, which has the possibility to use physical cameras and a better lighting model, as well model more of the DSLR intrinsic parameters.

## B.7 Acknowledgements

This work is funded by the LER project no. EUDP 2015-I under the Danish national EUDP programme. This funding is gratefully acknowledged.

## References

- [1] N. Statham, “Use of photogrammetry in video games: a historical overview,” *Games and Culture*, p. 1555412018786415, 2018.
- [2] Q. Péntek, S. Hein, A. Miernik, and A. Reiterer, “Image-based 3d surface approximation of the bladder using structure-from-motion for enhanced cystoscopy based on phantom data,” *Biomedical Engineering/Biomedizinische Technik*, vol. 63, no. 4, pp. 461–466, 2018.

## References

- [3] L. M. Galantucci, M. G. Guerra, and F. Lavecchia, "Photogrammetry applied to small and micro scaled objects: A review," in *International Conference on the Industry 4.0 model for Advanced Manufacturing*. Springer, 2018, pp. 57–77.
- [4] D. Pritchard, J. Sperner, S. Hoepner, and R. Tenschert, "Terrestrial laser scanning for heritage conservation: the cologne cathedral documentation project." *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 4, 2017.
- [5] E. R. Eiríksson, J. Wilm, D. B. Pedersen, and H. Aanaes, "Precision and accuracy parameters in structured light 3-d scanning," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, p. 7, 2016.
- [6] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer, "A survey of structure from motion\*." *Acta Numerica*, vol. 26, pp. 305–364, 2017.
- [7] Bentley, "Contextcapture," <https://www.bentley.com/en/products/brands/contextcapture>, 2016, accessed: 2020-06-15.
- [8] Agisoft, "Metashape (photoscan)," <http://www.agisoft.com/>, 2010, accessed: 2020-06-15.
- [9] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [10] C. Sweeney, T. Hollerer, and M. Turk, "Theia: A fast and scalable structure-from-motion library," in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015, pp. 693–696.
- [11] J. Schöning and G. Heidemann, "Evaluation of multi-view 3d reconstruction software," in *International conference on computer analysis of images and patterns*. Springer, 2015, pp. 450–461.
- [12] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, p. 78, 2017.
- [13] Autodesk, "Maya," <https://www.autodesk.eu/products/maya/overview>, 1998, accessed: 2018-11-10.
- [14] C. Group, "V-ray," <https://www.chaosgroup.com/>, 1997, accessed: 2018-11-10.
- [15] ISO12232, "Photography - digital still cameras - determination of exposure index, iso speed ratings, standard output sensitivity, and recommended exposure index," <https://www.iso.org/standard/37777.html>, 2006, accessed: 2018-11-05.

## References

- [16] D. A. Kerr, "Apex-additive system of photographic exposure," *Issue*, vol. 7, no. 2007.08, p. 04, 2007.
- [17] M. Born and E. Wolf, *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. Elsevier, 2013.
- [18] D. Girardeau-Montaut, "Cloudcompare-open source project," *Open-Source Project*, 2011.
- [19] I. Nikolov and C. Madsen, "Benchmarking close-range structure from motion 3d reconstruction software under varying capturing conditions," in *Euro-Mediterranean Conference*. Springer, 2016, pp. 15–26.

# Paper C

LiDAR-based 2D Localization and Mapping System  
using Ellipse Distance Correction Models for UAV  
Wind Turbine Blade Inspection

Ivan Nikolov and Claus Madsen

The paper has been published in the  
*Proceedings of 12th International Joint Conference on Computer Vision, Imaging  
and Computer Graphics Theory and Applications*, pp.418-425, 2017.

© 2017 SCITEPRESS Digital Library  
*The layout has been revised.*

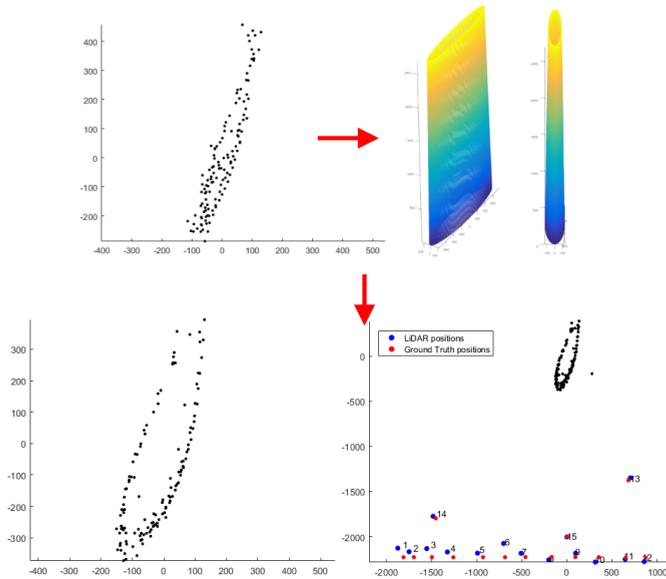
## Abstract

*The wind energy sector faces a constant need for annual inspections of wind turbine blades for damage, erosion and cracks. These inspections are an important part of the wind turbine life cycle and can be very costly and hazardous to specialists. This has led to the use of automated drone inspections and the need for accurate, robust and inexpensive systems for localization. Due to low geometrical features on the wind turbine blade, conventional SLAM algorithms have a limited use. We propose a cost-effective, easy to implement and extend system for on-site outdoor localization and mapping in low feature environment using the inexpensive RPLIDAR and an 9-DOF IMU. Our algorithm geometrically simplifies the wind turbine blade 2D cross-section to an elliptical model and uses it for distance and shape correction. We show that the proposed algorithm gives localization error between 1 and 20 cm depending on the position of the LiDAR compared to the blade and a maximum mapping error of 4 cm at distances between 1.5 and 3 meters from the blade. These results are satisfactory for positioning and capturing the overall shape of the blade.*

## C.1 Introduction

As robots and drones become widely used in different branches of the industry, a need for localization and mapping systems, which are able to work in non-ideal conditions arises. As the state of the art for drone localization, simultaneous localization and mapping (SLAM) is most widely used both from monocular [1] and stereo camera [2], LiDAR and laser scanners [3, 4]. A lot of SLAM algorithms for the outdoors are also used together with positioning from a GPS and orientation from an inertial measurement unit (IMU) [5, 6]. SLAM algorithms work best in environments, which are rich in geometric features and vary significantly. The precision of these algorithms significantly suffers when the environment is too featureless or when there are not enough points of interest, which requires the introduction of algorithm modifications like visual feature clustering [7] or combining camera and laser scanner depth data [8]. When the extracted points from laser scanner are not enough, due to sharp corners or thin surfaces, these modification cannot sustain a high enough performance. These problems are present when using drones in the wind energy sector for inspection of wind turbine blades. Normally wind turbine blades are at heights more than 100 meters and do not have any other landmarks around them. This, combined with the monochrome color of the blades and the lack of corners and other geometric features, makes autonomous localization of the drone and mapping the environment hard.

We propose a cost-effective and easy to implement solution for localization and mapping the overall shape of the blade (Figure C.1), designed to work in feature-poor wind turbine blade environments, with minimum



**Fig. C.1:** Our proposed localization and mapping solution using ellipse distance correction models.

prior information. Our system is made using the RPLIDAR scanner and the BNO055 9-DOF IMU for orientation. Our algorithm is based on the assumption that the blade is stopped in a vertical position, at either the top or bottom and it is the only thing that can be "seen" by the LiDAR. The algorithm uses prior information for the cross-section width and depth and it models the blade using primitive elliptical shapes and provides drone positioning in a much simpler way than conventional SLAM algorithms. The initial angle between the LiDAR and blade is found by using a 2D iterative closest point (ICP) algorithm and the angle heading is monitored and maintained using an IMU. Additionally we model the distance measurement and sampling errors of the RPLIDAR to boost the accuracy of the localization. We address the sunlight noise problem of the RPLIDAR with a hybrid hardware and software filtering solution. We perform initial ground based tests on the system to determine its accuracy and error rate by comparing with ground truth manually measured positions. Additionally we estimate to what extent the algorithm can capture the overall shape of the 2D cross-section of the blade, using a ground truth cross-section. Both tests demonstrate that the system performs well even with a small number of points present from the scanned area and can be used for drone localization and generation of initial sparse 2D point clouds of the shape of the blade.

## C.2 State of the Art

The use of drones for inspection of wind turbine blades has become a hot research topic in the fields of computer vision, automation and robotics. The limited number of points of interest, the featureless and monochrome shape of the wind turbine blades and the required working height, together with environment hazards like gusts of wind, blade swaying and weather changes make localization and mapping non-trivial problems. The research from [9] shows that inspection by images can suffer significantly if the drone's localization and stabilization systems cannot compensate for environment and weather changes. One way to tackle these problems is to have an initial fly-through, through which a occupancy map of the blade is created and which is then used for planning a flight path, as suggested by [10]. Another solution proposed by [11] can come from not only relying on a flying drone, but create a hybrid flying and climbing vehicle, which uses a combination of sonar, IMU and LiDAR system to orient itself. Using a dynamic flight path planning from a camera feed is another strategy employed by [12] and [13], combining it together with navigation sensors like an IMU, GPS, barometer, etc.

We propose a localization and shape mapping system, which uses a LiDAR scanner and a IMU, plus a simplification of the wind turbine blade to a elliptical model for distance and shape correction. Our algorithm is based on the assumption that only the blade is visible at any time and works even when as little as 2 or 3 points are obtained from the scanned surface. Before the correction is done, the data from the LiDAR is cleared of any sunlight noise readings and the calculated distance errors are removed. The estimated position of the drone is also filtered from noise to provide a stable localization.

## C.3 Methodology

The proposed system, given in Figure C.2, relies on a prior shape information for a proper positioning. The prior information in our case comes in the form of a model of the shape of the blade cross-section. Because finding a real world 3D model for each piece of the cross-section of the different kinds of blades in a non-trivial and almost impossible task, a simplified elliptic model is proposed. The elliptical model requires much less prior information and it gives centimeter precision results as shown in Section C.4. Stray sunlight in the sensor causes detection noise, which is filtered away using a combination of a hardware polarization and software k-nearest neighbors and line filters, before the LiDAR raw data is used.

As an initial step, before the main localization algorithm is started, the

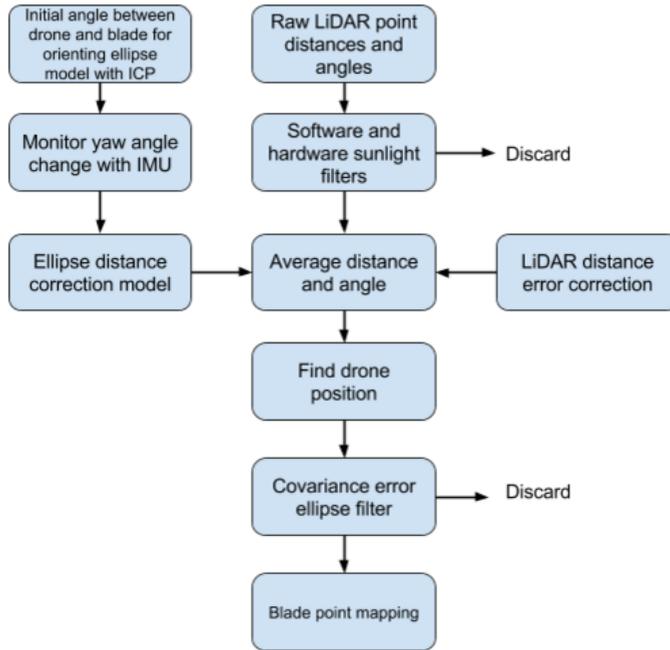


Fig. C.2: Algorithm and work pipeline overview. From raw LiDAR data and IMU orientation, to localization of the LiDAR compared to the blade and mapping the points of the blade cross-section

angle between the blade and drone is calculated using readings from a low-cost LiDAR from [14], shown in Figure C.3a and an 2D ICP algorithm, as described in Subsection C.3.3. The angle is used to rotate the ellipse model to be the same direction as the blade and is then maintained with the readings from the IMU from [15], shown in Figure C.3b. It is observed that the LiDAR exhibits errors in its readings which change with the distance from the measured surface. This error is compensated for by comparing measurements from the LiDAR and a ground truth readings DISTO laser distance scanner. The differences are calculated, interpolated and smoothed out using a cubic spline and used as distance corrections.

Once the data is clear of noise, corrected from the distance errors and rotated properly, the ellipse distance correction model (EDC model) as seen in Subsection C.3.2, together with the angle correction from the IMU, are applied. The localized drone position is filtered to remove false movement readings, demonstrated in Subsection C.3.4 and the blade points are mapped accordingly.



Fig. C.3: The hardware used for the proposed system - the RPLIDAR and the BNO055 9-DOF IMU

### C.3.1 From Corrected LiDAR Data to Initial Drone Localization

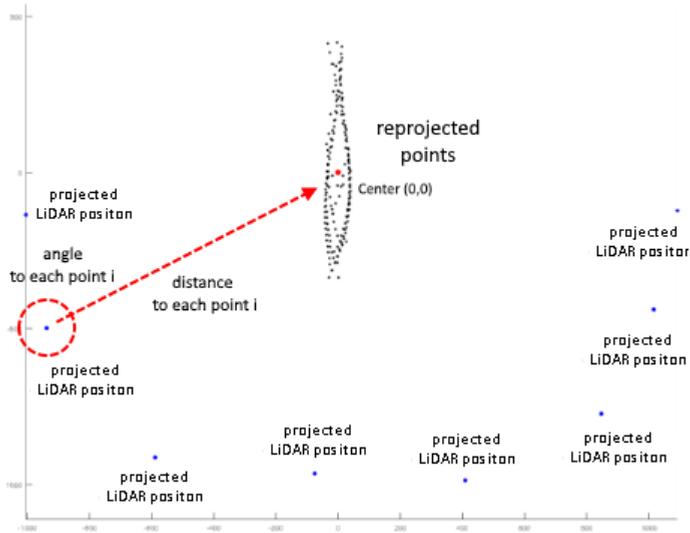
The corrected output data from the RPLIDAR comes in the form of angles between  $0$  and  $2\pi$  for each detected object point, together with the distance to these points. To position the data for each LiDAR reading in a unified coordinate system the data needs to be first transformed from a polar coordinate system to a Cartesian one. We assume that only the wind turbine blade is seen by the LiDAR at any given time, so the origin of the new Cartesian coordinate system is chosen to be the center of the scanned blade. Formula C.1 is used for transforming the coordinates.

$$\begin{aligned} x_d &= 0 - D_{mean} \cdot \sin(\alpha_{mean}) \\ y_d &= 0 - D_{mean} \cdot \cos(\alpha_{mean}) \end{aligned} \quad (C.1)$$

Where the  $x_d$  and  $y_d$  are the drone coordinates in the new coordinate system,  $D_{mean}$  and  $\alpha_{mean}$  are the mean distance and angle calculated from all detected points in a 360 degree reading and  $(0,0)$  is chosen as the center of the polar coordinate system. Once the LiDAR is self localized, compared to the blade, the LiDAR's position is used as a new center of a polar coordinate system to find the blade's previously detected points. The points are calculated in a Cartesian coordinate system using Formula C.2.

$$\begin{aligned} x_i &= x_d + D_i \cdot \sin(\alpha_i) \\ y_i &= y_d + D_i \cdot \cos(\alpha_i) \end{aligned} \quad (C.2)$$

Here  $x_i$  and  $y_i$  are the coordinates of each of the data points in one reading and  $D_i$  and  $\alpha_i$  are respectively their distance and angle. The centers used for projecting the blade points are the LiDAR's coordinates  $x_d$  and  $y_d$ , calculated in Formula C.1. The whole process is visualized in Figure C.4. The initial naive approach to localization and mapping yields some problems, such that the whole cross-section needs to be opened up. Additional processing is required to get the correct cross-section mapping and position localization. This is why we introduce the ellipse distance correction model (EDC model).



**Fig. C.4:** Initial drone localization. The corrected LiDAR data is transformed into a Cartesian coordinate system and the mean angle and distance are used to self-position the LiDAR compared to the blade. Once these positions are found the detected blade points are reprojected from the LiDAR position centers. Eight LiDAR positions and the "seen" blade points are plotted in the Figure

### C.3.2 Ellipse Distance Correction Model (EDC model)

Each reading of the LiDAR to the wind turbine blade gives the distance from the seen blade part to the LiDAR. This model of positioning does not take into account the full shape and volume of the object that it is building a profile of. Because each of the LiDAR positions is given from the center of the coordinate system (0,0), these positions are not correct. In real life the LiDAR does not measure the distance from the drone to the blade center, but to the exterior shell/body of the blade. The distance from the center of the blade to the exterior points that the LiDAR measures is not known to the LiDAR. This discrepancy leads to improper modeling of the surface shape of

the blade and incorrect drone position calculation.

To be able to capture the proper shape of the blade profile and the proper LiDAR positions, an initial guess of the blade form is needed. The wing of a wind turbine has a very specific and hard to model shape, which changes with height. Each blade manufacturer has different specifications, numbering and shape mold. This information is not readily available. This requires a simplified model, which with some approximation can be used as a substitute to the blade shape. This model can be further adjusted using known blade shapes. This also requires two additional assumptions to be done:

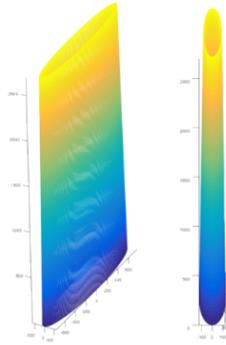
- The LiDAR sees only the wind turbine blade;
- The wind turbine blade is stopped;
- The blade's cross-section width and depth are known;
- The vertical position of the drone is known.

An ellipse is suggested as such a simple geometrical shape, as it closely resembles the the blade cross-section, especially in the tip. To model properly the blade's size for the different heights, an ellipse shape is calculated for each height slice of the blade. The ellipse points are calculated using the Formula C.3, where  $r_1$  and  $r_2$  are the two radii of the ellipse, which are selected based on the width and depth of the cross-section at the particular blade height and the angle  $\beta \in [0 : 2\pi]$ .

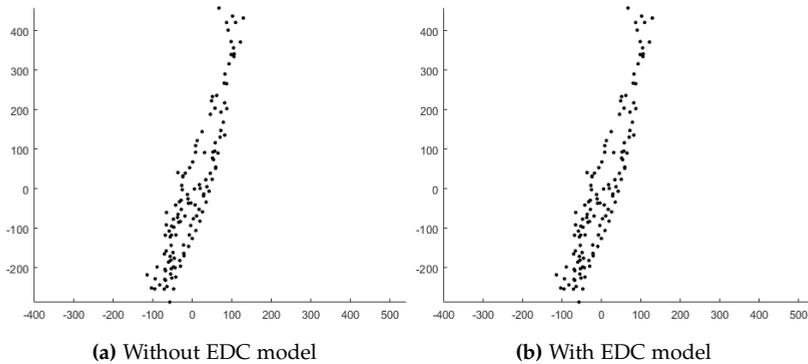
$$\begin{aligned} x_e &= 0 - r_1 \cdot \sin(\beta) \\ y_e &= 0 - r_2 \cdot \cos(\beta) \end{aligned} \tag{C.3}$$

A full model of the blade is created and is saved in a table, for an easy access depending on the height of the drone, compared to the blade. Multiple ellipse models can be created for different blade dimensions and manufacturer models. A full 3D model can be seen in Figure C.5.

For each point on the ellipse the radius from the center to the point is calculated, as well as the angle on a circle with a center in (0,0). Each point on the ellipse is positioned in the same coordinate system as the points from the LiDAR. This way when the LiDAR position is calculated using the mean distance, the distance of the point on the ellipse with the closest angle to the mean angle of the LiDAR is added to it. This effectively pushes outwards the mean points and gives an initial guess on the shape of the blade. The difference can be seen when plotting the blade surface points before and after the ellipse correction, as seen in Figure C.6 The correction is most noticeable on the points on the tip of the blade. The whole profile is opened up, representing the blade cross-section more closely.



**Fig. C.5:** Ellipse model for correcting the LiDAR's position. The 3D model is created with interpolation of the two radii of the ellipse cross-section in Z direction, following the notion that wind turbine blades become smaller with height.



**Fig. C.6:** Example mapping of the blade before and after the EDC model. The points' positions are corrected and the whole profile is opened up, especially the blade edge points.

Another problem becomes evident when using the EDC model - the ellipse needs to be initially oriented the same way as the blade profile. The rotation of the blade is initially unknown and needs to be determined before the correction can be applied to the localization and mapping.

### C.3.3 Calculating Initial Angle Between Drone and Blade

The drone's yaw orientation is taken from the IMU unit. The angle between the detected blade profile from the LiDAR and the drone is calculated. The initial angle is used to first orient the ellipse compared to the blade's orientation for the EDC model and also for keeping the mapped positions at the same heading. The change in the angle of the wind turbine blade is deemed

too small, as it will be stopped from moving or rotating for the duration of the drone flight.

Once a reading from the LiDAR with enough points is registered the angle between the blade and LiDAR can be calculated using ICP. Heuristically it is determined that at least 8 points are required in one reading for the ICP registration to be successful. The sides of the blade are better for registration than the edge, as they contain more points, which are spread more evenly. Because the success of the ICP depends on the quantity and quality of the registration points, a cubic spline is approximated to the scanned points. This way the scanned points are interpolated and any small noise position errors are smoothed over. A 2D ICP algorithm by [16] is used to register the LiDAR points to the prior ellipse model. A rigid registration is performed with only rotation and translation, as the ellipse is properly sized beforehand, using the initial information for the width and depth of the blade profile.

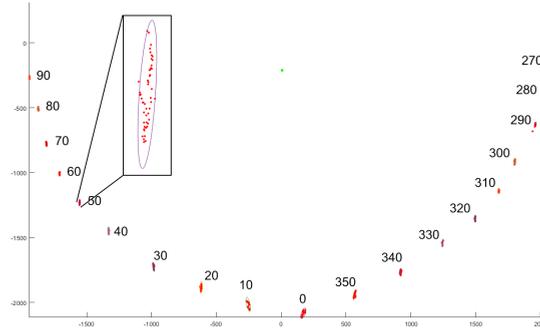
Once the registration is completed, the rotation angle is used to rotate the EDC model. The angle is also added to the yaw angle measured by the IMU for use with the drone's yaw holding.

#### **C.3.4 LiDAR position filtering**

The calculated LiDAR position compared to the blade fluctuates in time, because of imperfections of the LiDAR, the varying scanning angles, as well as the amount of the blade profile seen by the system. Keeping to the theme of simple solutions throughout the paper, a covariance ellipse error filtering is chosen, as shown by [17]. This requires the calculation of a covariance ellipse for each position. Readings which are inside the radius of the ellipse will be deemed noise and will be disregarded, while the readings outside of it will be considered as valid readings from the movement of the drone with the LiDAR.

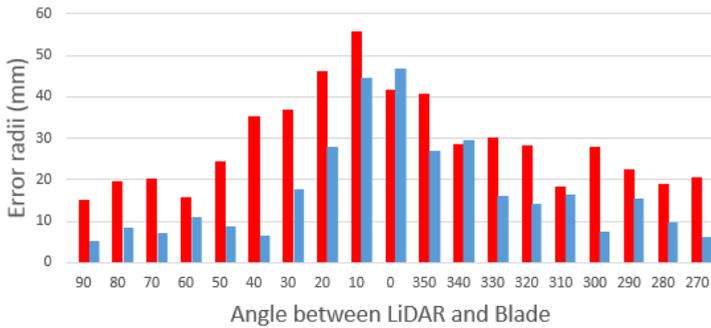
It is observed that the variance fluctuation changes depending on the position of the LiDAR compared to the blade, as the narrower the blade becomes, the less points there are to detect. Additionally the steeper the seen surface of the blade is from the position of the LiDAR, the bigger positioning errors are resultant from the varying scanning angle of the LiDAR. To calculate the covariance error ellipse in the needed positions, a semi-circle around the blade edge is chosen. Fifty measurements with the LiDAR are taken in intervals of 10 degrees. A plot of the measured positions, angles at each position, together with a "zoomed in" error ellipse are given in Figure C.7. This angle orientation is taken from the scanning orientation of the RPLIDAR.

The error ellipse radii are interpolated for each degree in the semi-circle. Before each calculated position is accepted as a proper one, it is tested against the accumulated mean of known positions. If it is inside the ellipse radius for the current LiDAR position angle, with center the accumulated mean of



**Fig. C.7:** Positioning error ellipse created from 50 RPLIDAR measurements. The LiDAR is positioned on each position and is not moved, while the measurements are taken. The readings from a semi-circle around the blade are used to create a filter for removing position noise. Bigger position deviations are present at the direction the blade is oriented.

previous know positions, it is discarded. Figure C.8 gives the largest and smallest radii of the error ellipse at each measured angle.



**Fig. C.8:** Position filtering small (blue) and large (red) radii(mm) for measurements in 180 degrees around the wind turbine blade. The radii are getting smaller the farther from the blade’s edge the measurement is taken.

## C.4 System Tests and Results

### C.4.1 Localization Test

The first test aims to measure the localization accuracy of finding the position of the LiDAR compared to the blade. As the system is still in the prototyping stages, the measurements are done on the ground. A number of ground truth points are measured around the blade profile, around the blade’s leading edge between 0.5m and 3.5m. Two sets of fifteen positions are selected.

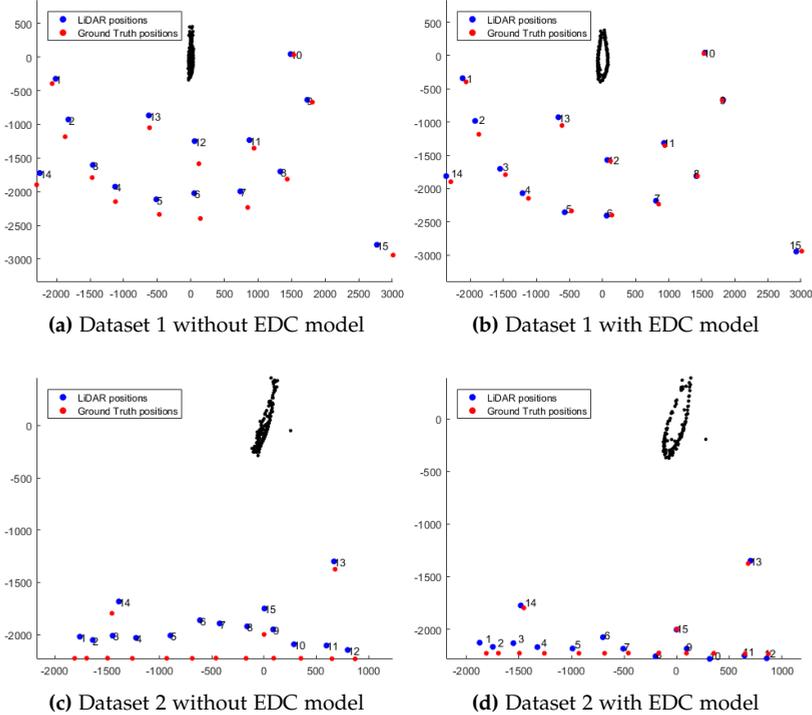
#### C.4. System Tests and Results

The first set is with positions arranged in a circle and the second - in a line. These are selected, to demonstrate the system’s performance with different scan patterns. Both sets contain outliers positions to verify the accuracy of the system in harder localization points. The LiDAR is positioned on each of the ground truth points and the measured position is computed. Fifty readings are taken so any noisy measurements can be filtered away. The Euclidean distance between each of the ground truth positions and measured LiDAR self positioning is computed, both with and without the ellipse distance correction model. The resultant error values from the first and second position datasets are given in Table C.1 and the plotted LiDAR and blade points are given in Figure C.9.

**Table C.1:** Results from the first and second set of positions around the blade. The distance error(*mm*) between the ground truth position and the one measured by the LiDAR with the correction algorithm  $E_{algorithm}$  and with only the raw data  $E_{raw}$

Pos	Dataset 1		Dataset 2	
	$E_{raw}$	$E_{algorithm}$	$E_{raw}$	$E_{algorithm}$
1	88.362	76.005	213.102	119.219
2	259.963	210.010	183.806	79.882
3	187.537	119.011	221.196	111.193
4	221.324	118.634	199.417	88.595
5	228.452	104.669	221.512	75.396
6	386.517	81.497	372.165	153.788
7	263.294	64.543	333.901	64.492
8	152.989	16.175	306.109	41.510
9	84.319	12.198	276.031	43.078
10	47.174	18.259	150.436	66.077
11	138.618	42.206	136.805	18.486
12	341.866	51.908	110.823	45.537
13	183.159	135.091	75.766	34.394
14	180.572	110.274	131.246	36.796
15	285.282	84.674	248.219	4.591
<b>Avg</b>	<b>203.295</b>	<b>83.010</b>	<b>212.036</b>	<b>65.536</b>

The results show that the system can provide a stable position estimate for where the drone is compared to the blade. The biggest differences between the measured and ground truth points come from positions where just a small amount of points from the surface can be seen. In addition positions which are too far away from the blade surface exhibit both lower precision and accuracy, because the not enough points are sampled by the LiDAR, which makes faraway point angle and distance estimation jump around too much. Some larger errors are observed in the first dataset, which can be also



**Fig. C.9:** Localization test using the first (C.9a and C.9b) and second (C.9c and C.9d) position datasets. The ground truth data is shown with red dots and the LiDAR calculated positions with blue dots. The mapped blade points are also plotted, for easier orientation

explained with the problems with the yaw hold and small changes in the rotation of the LiDAR. In all positions, the introduction of the EDC model is shown to dramatically increase the accuracy of the localization.

## C.4.2 Mapping Test

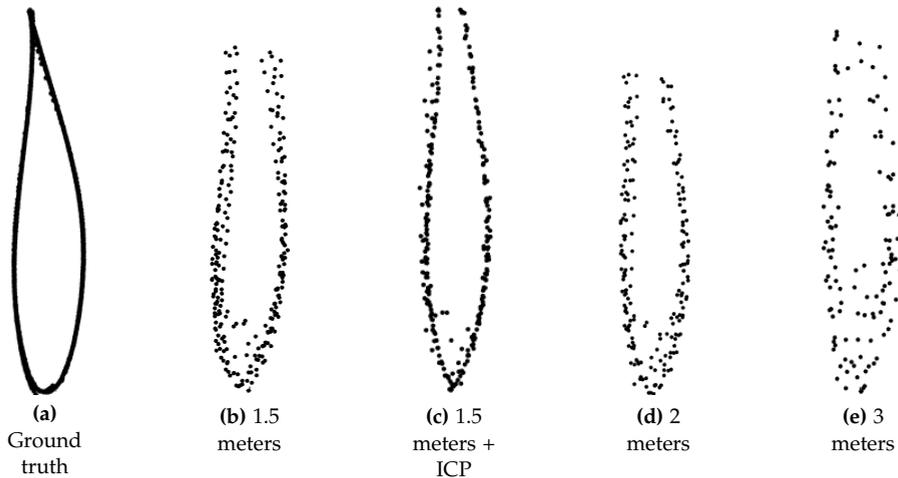
For testing the mapping capabilities of the LiDAR system, the blade profile is scanned with a Faro laser scanner and a high quality point cloud is created. A 2D cross-cut of the point cloud is then used as a ground truth, when comparing with the mapped blade point positions. In this test the LiDAR is placed in positions, forming a semicircle around the blade with three radii - 1.5 m, 2 m and 3 m, for testing the mapping capabilities from different distances. Additionally a post-processed mapping from 1.5 m, which has been registered to the ellipse prior model using the ICP algorithm is given, to demonstrate one way that the proposed algorithm can be extended to boost accuracy. The signed distances between each point of the two 2D point clouds is calculated.

#### C.4. System Tests and Results

**Table C.2:** Mean value ( $\mu$ ) in  $mm$ , standard deviation ( $\sigma$ ) in  $mm^2$ , minimum ( $d_{min}$ ) and maximum ( $d_{max}$ ) in  $mm$  of the absolute distance metric between the ground truth and the LiDAR mapped points from three distances to the blade ( $D_b$ ) - 1.5, 2 and 3 meters

$D_b$	$\mu$	$\sigma$	$d_{min}$	$d_{max}$
1.5	10.931	7.322	0.363	38.130
1.5 + ICP	9.323	6.920	0.240	31.038
2	14.891	9.185	0.624	38.395
3	15.645	9.234	1.306	39.296

This is done using the free open source software CloudCompare [18]. From the distances, the mean and standard deviation of the distance distribution are calculated. The mean and standard deviation are given in Table C.2 and the mapped blade surface points are given in Figure C.10.



**Fig. C.10:** Mapped blade surface from the LiDAR, using the EDC model, together with the ground truth Faro scan and the post-processed 1.5 meter mapping using the ICP algorithm together with the ellipse prior model. The 2D point clouds get sparser and noisier the farther away the LiDAR is.

The difference between the ellipse correction model and the blade cross-section becomes larger the farther away from the edge it goes, because the scanning is done only in a 180 degree semi-circle around the blade and details of the back of the blade are sparse. The front of the blade in the LiDAR scans shows a higher standard deviation and some noisy points, because of the small amount of points seen from those angles. Even with these problems and the relatively small sampling density of the RPLIDAR, the proposed algorithm manages to restore the shape of the blade with centimeter accuracy.

If a registration is done, the results get closer to the ground truth shape. This shows that the algorithm can be used as a proper substitute to SLAM.

## C.5 Conclusion and Future Work

We propose a low-cost, easy to implement drone localization and mapping system for wind turbine blades, using a cheap commercial LiDAR and a off-the-shelf IMU. Our system uses prior shape information in the form of the ellipse distance correction model. It requires minimum prior information, it is computationally fast, simpler to implement than conventional SLAM, it can be easily extended and refined with additional training and provides satisfactory results. We demonstrate through ground based localization and a mapping tests that the system can self position and obtain mapping of the blade cross-cut with centimeter accuracy. In addition we propose a filter for removing noisy position information. Our algorithm also removes distance measurement errors and direct sunlight noise, so it can be used both outdoors and indoors.

As an extension of our work we propose an in-depth test of our system and algorithm against the state of the art SLAM algorithms performed on blade profiles, to verify the calculated accuracy of the system. Testing the algorithm using a "default" blade shape, which will better resemble the scanned blades, as a distance correction model is also suggested, as well as further adjusting it using training sets.

## Acknowledgements

This work is funded by the LER project no. EUDP 2015-I under the Danish national EUDP programme. This funding is gratefully acknowledged.

## References

- [1] R. Mur-Artal, J. Montiel, and J. D. Tardós, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [2] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the rgb-d slam system," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1691–1696.
- [3] C. Berger, "Toward rich geometric map for slam: Online detection of planes in 2d lidar," *Journal of Automation Mobile Robotics and Intelligent Systems*, vol. 7, 2013.

## References

- [4] R. Zlot and M. Bosse, "Efficient large-scale 3d mobile mapping and surface reconstruction of an underground mine," in *Field and service robotics*. Springer, 2014, pp. 479–493.
- [5] J. Elseberg, D. Borrmann, and A. Nüchter, "6dof semi-rigid slam for mobile scanning," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 1865–1870.
- [6] D. P. Shepard and T. E. Humphreys, "High-precision globally-referenced position and attitude via a fusion of visual slam, carrier-phase-based gps, and inertial measurements," in *2014 IEEE/ION Position, Location and Navigation Symposium-PLANS 2014*. IEEE, 2014, pp. 1309–1328.
- [7] P. L. Negre, F. Bonin-Font, and G. Oliver, "Cluster-based loop closing detection for underwater slam in feature-poor regions," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 2589–2595.
- [8] T. Oh, H. Kim, K. Jung, and H. Myung, "Graph-based slam approach for environments with laser scan ambiguity," in *Ubiquitous Robots and Ambient Intelligence (URAI), 2015 12th International Conference on*. IEEE, 2015, pp. 139–141.
- [9] G. Morgenthal and N. Hallermann, "Quality assessment of unmanned aerial vehicle (uav) based visual inspection of structures," *Advances in Structural Engineering*, vol. 17, no. 3, pp. 289–302, 2014.
- [10] B. E. Schäfer, D. Picchi, T. Engelhardt, and D. Abel, "Multicopter unmanned aerial vehicle for automated inspection of wind turbines," in *2016 24th Mediterranean Conference on Control and Automation (MED)*. IEEE, 2016, pp. 244–249.
- [11] S. Jung, J.-U. Shin, W. Myeong, and H. Myung, "Mechanism and system design of mav (micro aerial vehicle)-type wall-climbing robot for inspection of wind blades and non-flat surfaces," in *Control, Automation and Systems (ICCAS), 2015 15th International Conference on*. IEEE, 2015, pp. 1757–1761.
- [12] C. Galleguillos, A. Zorrilla, A. Jimenez, L. Diaz, Á. Montiano, M. Barroso, A. Viguria, and F. Lasagni, "Thermographic non-destructive inspection of wind turbine blades using unmanned aerial systems," *Plastics, Rubber and Composites*, 2015.
- [13] M. Cordero, M. Trujillo, J. Ruiz, A. Jimenez, L. Diaz, and A. Viguria, "Flexible framework for the development of versatile mav systems for multi-disciplinary applications," in *IMAV 2014: International Micro Air*

## References

*Vehicle Conference and Competition 2014, Delft, The Netherlands, August 12-15, 2014.* Delft University of Technology, 2014.

- [14] Slamtec, “Rplidar,” <http://www.slamtec.com/en>, 2010, accessed: 2018-09-06.
- [15] Adafruit, “Adafruit bno055 9-dof imu,” <https://learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor>, 2012, accessed: 2016-09-15.
- [16] P. Bergström and O. Edlund, “Robust registration of point sets using iteratively reweighted least squares,” *Computational Optimization and Applications*, vol. 58, no. 3, pp. 543–561, 2014.
- [17] M. Friendly, *SAS system for statistical graphics*. SAS Publishing, 1991.
- [18] D. Girardeau-Montaut, “Cloudcompare-open source project,” *Open-Source Project*, 2011.

# Paper D

## How Image Capturing Setups Influence the Quality of SfM Reconstructions for Wind Turbine Blade Inspection

Ivan Nikolov, Emil Krog Kruse and Claus Madsen

The paper has been accepted to  
*SPIE Future Sensing Technologies*

*The layout has been revised.*

### Abstract

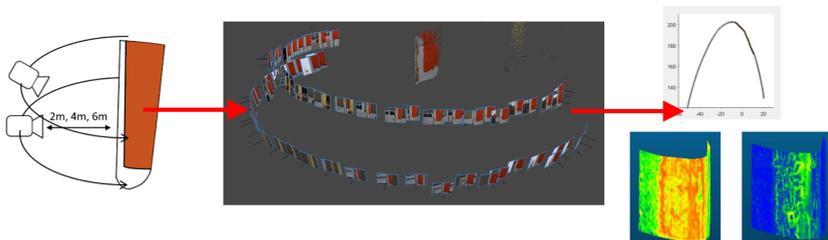
*Increased leading edge roughness (LER) is one of the main causes for wind turbine blade performance degradation. To ensure a consistently high energy output, the surface erosion of wind turbine blades, needs to be monitored regularly, so preventive measurements can be done. Capturing 3D surface data is becoming a more and more popular way to extract and quantify roughness on a micro level. In this paper we want to test the possibility of using Structure from Motion (SfM) 3D reconstructions for extracting surface roughness information from wind turbine blades. We test various capturing scenarios with varying horizontal and vertical image overlap, as well as varying distances to the blade, using a real blade in outdoor conditions. We analyze the quality of the reconstructions and provide a benchmark, as well as guidelines to what are the best possible capturing conditions for ensuring a high quality and noise free 3D surface results.*

### D.1 Introduction

Maximizing wind turbine performance is one of the main goal of the wind energy production industry. For achieving this a large part of the research goes into the design [1], optimization [2] and ensuring the longevity [3] of wind turbine blades. A big part of ensuring this longevity, falls on ensuring a constant monitoring of the surface health of wind turbine blades. The presence of high leading edge roughness (LER) can be detrimental to the energy production of the wind turbine blade, with loses between 2% and 5% [4] for smaller surface imperfections and 8% and 25% for more severe blade damage [5]. An important fact to consider is that larger blade damages start off from imperfections in the blade surface, because of manufacturing errors, dirt and debris buildup or changing weather [6]. These surface imperfections grow and worsen with time, leading to coating delamination and the depreciation of the wind turbine blade. To combat these more severe outcomes, early onset LER needs to be properly monitored.

Surface monitoring is done in a number of ways - by manual inspections of the wind turbine blades by expert engineers directly on-site, by sub-surface inspections using ultra sound [7] and thermal cameras [8] and by capturing the exterior surface with images, for later in-depth analysis [9]. A problem with these methods is that, detecting small surface imperfections cannot always be possible, because of the color and geometrical profile of LER. For these cases capturing the 3D surface can provide this information. There exist a number of hardware solutions for capturing 3D information from the environment - laser scanners [10], stereo cameras [11], structured light cameras [12], time-of-flight cameras [13]. A problem with most of these is that they either do not provide the robustness for being use in an outdoor, hard

to reach environments like wind turbines or do not provide the necessary scanning resolution to capture small enough details from the surfaces.



**Fig. D.1:** Example of the experimental setup, used to test the resultant SfM reconstructions, created from different distances and angular coverages. We analyse both the overall blade edge shape, as well as the visual and geometrical 3D profile of the reconstructions

Another possible solution is Structure from Motion (SfM), which combines the needed robustness and resolution requirements. The research by [14] has proven that SfM using high enough resolution cameras, can achieve sub-millimeter accuracy, matching microscopy results, if optimal capturing conditions are provided. In this paper we focus on benchmarking what constitutes these optimal conditions. We propose a number of testing scenarios for reconstructing the surface of a blade mock-up in outdoor conditions, using varying horizontal and vertical angular coverage and the distance between the blade and the camera. The blade mock-up is a part of a real turbine blade, that has been sandblasted to provide different levels of LER. The resultant SfM reconstructions are then analyzed to evaluate both the quality of captured shape of the blade, as well as the surface roughness. The results are then used as guidelines for capturing conditions give the best possible results. We show that adding more images can be beneficial for making the outcome of the reconstruction more robust to failing, but it also runs the risk of introduction of noise and shape errors. We also show that surface roughness detail degrades with distance, but more angular coverage can alleviate the deterioration. An example of the testing setup is shown in Figure D.1.

## D.2 State of the Art

Structure from Motion has been used for surface inspection and capturing roughness details in many industrial contexts. It is widely used in infrastructure and tunnel and railway inspection [15], [16], road analysis [17], geomorphological analysis [18], forestry [19] and cultural heritage [20]. Its use in wind turbine blade inspection [21], [22] has also started to be considered for analysing the depth of detected surface damages.

### D.3. Methodology

To produce correct representations of the imaged surfaces or objects, the quality of the reconstructions need to be able to represent small details and be free of noise and geometrical errors. Because of its reliance on 2D image data as input, SfM is dependent on the quality of the used camera sensor, lens and settings, as well on the capturing setup. Many factors go into the capturing setup: amount of images taken, the horizontal and vertical angular coverage from images, the distance from the subject to the camera, the illumination and the environment, as well as on the surface of the scanned object. A number of benchmarks have been created to test different aspect of the SfM capturing process for close-range and large-scale 3D reconstructions [23, 24]. Some of them compare the between SfM and other 3D capturing methods [25], while other focus on testing specific SfM solutions, under varying capturing conditions [26, 27].

For our current paper we take inspiration by the methods and results presented by [28, 29] and we continue the work on SfM for wind turbine blade reconstruction [14, 22], by verifying how the results degrade with the introduction of less optimal capturing setups. It has been shown that the main reasons for lower quality reconstructions is the lack of overlap [30], angular coverage [31] and larger distances between camera and subject [32]. In this paper we focus on these three factors. We test the captured data both using 2D metrics for analyzing the quality of the captured blade shape, as well as utilizing 3D shape and color intensity metrics for determining how the surface roughness representation changes.

## D.3 Methodology

### D.3.1 Structure from Motion Overview

Structure from Motion is part of the family of photogrammetry type of reconstruction algorithms, used for capturing measurement data through the use of images. The algorithm relies on reconstructing the 3D shape of objects or surfaces through multi-view image capture, achieved through positional and angular coverage. The taken images need to "see", the surface from different perspectives, with an appropriate overlap between them [33], of at least 70% to 80%. Keypoints and surface features, like edges, corners or ridges are extracted from each image and matched between them. Traditional computer vision algorithms like SIFT [34] or ORB [35] are used for that. Outlier and noise ones are then removed using methods like RANSAC [36]. The left matched features are then combined with the intrinsic camera parameters, taken from EXIF data and used to triangulate the camera positions, from which the images were taken, together with creating a sparse point cloud. This is done iteratively for all present images, until a rough estimate of the



**Fig. D.2:** The wind turbine segment used for the research in the paper, mounted horizontally on a gantry. The blade can sway in the wind unobstructed, for a more realistic image capture setup.

captured scene is created. This rough 3D estimate is then refined using algorithms like bundle adjustment [37], for minimizing the reprojection errors, between the calculated and predicted sparse points. Next a dense point cloud is calculated from the know camera positions and sparse points, using reprojection and interpolation of the known information. Finally, if needed 3D meshes can also be created from the dense point clouds, as well as textures, using the image information.

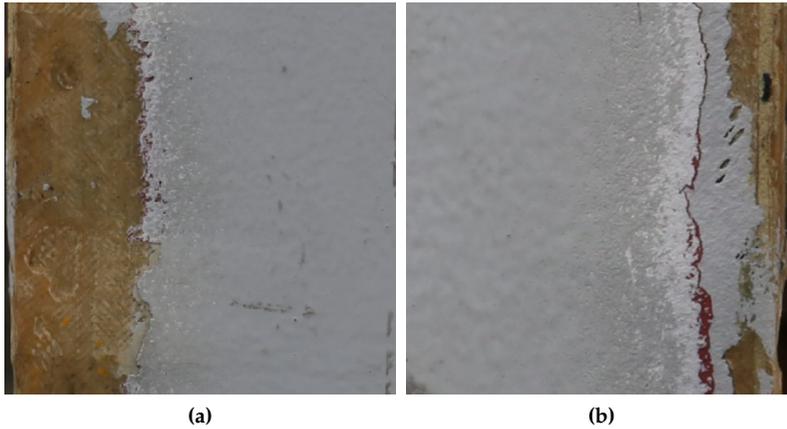
From this it can be seen that the quality of the output of SfM, highly depends on the quality and positions of the captured images. If not enough information can be extracted from them, this can lead to errors and noise in the reconstructions or even to failure of the whole processing pipeline. To test what effects, which different image capturing setups can have on the reconstructions of wind turbine blade surfaces, in the next subsection, we will present the capturing methodology used in the paper.

### D.3.2 Experimental Setup

For making the experiments a wind turbine blade segment from a decommissioned pitch-regulated wind turbine is selected. The segment is mounted vertically on a gantry, through the use of chains for easier access and to ensure that the blade still sways in a natural way, for a image capturing setup closer to real life. The blade segment can be seen in Figure D.2. Before mounting

### D.3. Methodology

the blade segment it has been sandblasted, to simulate the different levels of wear and tear, present on an in-use wind turbine. This produces variable surface roughness and damaged areas between 1-3mm along the leading edge of the blade. It has been demonstrated that surface roughness above 1mm start to introduce efficiency loss to the energy production of wind turbines [38], so the created areas are deemed sufficient for producing real world results. A close-up of the area of the blade that is imaged and reconstructed is shown in Figure D.3



**Fig. D.3:** Parts of the blade leading edge, which are selected for 3D reconstruction, containing both clean and damaged regions

For the capturing setup a Canon 6D camera is chosen, together with a variable focus 24-105mm lens (Canon EF 24-105mm f/4L IS II USM), set to its maximum 105mm mark. The resolution of the camera is set to 5472x3648. The image capture mode is set to manual, for more control over all the settings and for ensuring that the exposure of the images does not change in a dynamic outdoor environment. Camera settings of aperture, shutter speed and ISO are set to their appropriate settings.

For testing how the reconstruction output depends on the capturing setup, three main factors are evaluated:

- Distance from the camera to the blade surface - 2m, 4m and 6m
- Horizontal angular coverage - number of horizontal images per vertical band - 9, 17 or 33
- Vertical angular coverage - one or two vertical bands, larger angular coverage.

The first factor is the distance from the camera to the blade surface. This determines the resolution, with which the surface will be captured or how many pixels would represent each millimeter of the surface. To calculate this the field of view (FOV) and the resolution of the camera are needed, as shown in Equation D.1, where  $w_{im}$  is the horizontal resolution. For calculating the FOV, Equation D.2 can be used, where  $D_s$  is the distance from the camera to the subject and AOV is the angle of view of the camera. The AOV can be calculated using Equation D.3, there  $s_w$  is the width of the sensor of the camera, but it can also be the height or diagonal, depending on which angle needs to be calculated and  $f$  is the focal length of the camera.

$$PixelPerMM = \frac{w_{im}}{FOV} \quad (D.1)$$

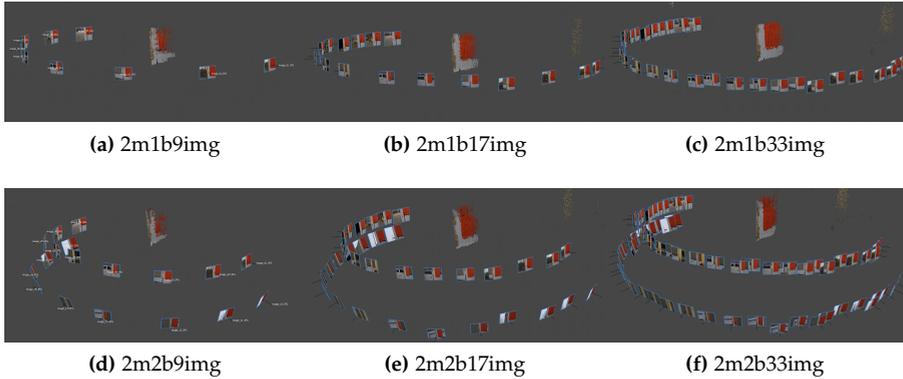
$$FOV = 2 \tan\left(\frac{AOV}{2}\right) D_s \quad (D.2)$$

$$AOV = 2 \arctan\left(\frac{s_w}{2f}\right) \quad (D.3)$$

For this experiment the focal length is set to 105mm, so maximum zoom level can be achieved and the only thing that is changed is the distance from the camera to the blade. All the images of the blade surfaces are taken in semi-circular bands around it, as this capturing setup has been proven to produce good results [28]. The lower bound of 2m is selected, as it is deemed that in a real world capturing scenario a unmanned aerial vehicle, would not be able to safely get closer to a wind turbine blade. The higher bound of 6m is restricted, by the size of the capturing location. Example of all the image capturing setups for 2m distance are given in Figure D.4.

The factor of horizontal angular coverage, determines how much of the wind turbine blade's surface is shared between images. This can influence the quality of the feature extraction and matching step of SfM. Images with insufficient horizontal angular coverage cannot produce enough matched features and can tend to produce incorrect matches, this can lead to failure of the initial triangulation or to incorrectly calculated camera positions. In addition, because of the shape of the wind turbine blade - a very narrow edge and two flatter, but still curved sides, the horizontal angular coverage can be even more important to ensure that a enough features can be matched between the two sides of the blade. This has proven a known problem when capturing thin or very curved surface [39], where a distinct lack of matched feature points is observed. For testing this 33 images were taken in a semi-circular band around the blade for each capturing distance, keeping the same radius. Later for producing the reconstructions, two additional datasets were constructed from these 33 images, by sampling every second and every fourth image, to create respectively a 17 and 9 image dataset.

### D.3. Methodology



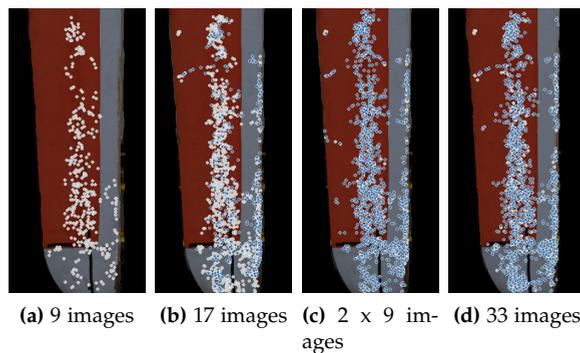
**Fig. D.4:** Reconstructed capturing setups for the 2m distance case for both 1 and 2 vertical bands and 9, 17 and 33 images per band. The capturing setups for the other two distance cases are setup the same. The blades follow a naming convention of - [distance], [number of bands], [number of images].

In contrast to the horizontal angular coverage, which is ensured by providing enough images, the vertical angular coverage, is made harder. Again because of the shape of the blade, capturing images of its surface from different horizontal angular views is made harder, and having such views is proven to positively impact the resultant reconstructions [29]. To test this an image dataset of 33 images in one horizontal band and a image dataset of 66 images in two horizontal bands were created. For the second band the camera was moved moved high enough to ensure a 15 degree angle difference between the two bands for each of the distances from the blade.

The 3 distance variations, 3 horizontal angular coverage variations and 2 vertical angular coverage variations, were combined to create 18 different datasets. Because the gantry with the blade was positioned in a relatively cluttered environment, it was decided to mask everything, but the blade in all images before using them for 3D reconstruction. This was done, so the SfM software, would not use information from the background to help with the reconstruction of the blade. This would be case in a real world capturing scenario, where the background would be either the featureless tower, the sky or the ground, which would also be far away and out of focus. Each of the datasets is then used as an input for Agisoft Metashape [40] and the SfM reconstruction of each is used in the Results section D.4.

## D.4 Results

All 18 datasets were produced by Metashape. From them three datasets could not be successfully reconstructed. The 6 meter, 1 vertical band, 9 and 17 image datasets failed on the initial camera and sparse point cloud calculation, while the 6 meter 2 vertical bands, 18 image dataset was reconstructed, but the output was incorrectly triangulated resulting in twisted and malformed results, not representative of the wind turbine blade surface and unusable for inspection. By analysing the found and matched features for these failed reconstructions and comparing them to a successful one, we can see that for the first two cases almost no features that were found could be matched between the images, while in the third case, the matched features were not enough. The result found and matched features one of the images are presented in Figure D.5, where the white dots represent found features and the blue ones represent matched features. Figure D.5d shows the found and matched features, from the dataset using 33 images, which was successfully reconstructed.

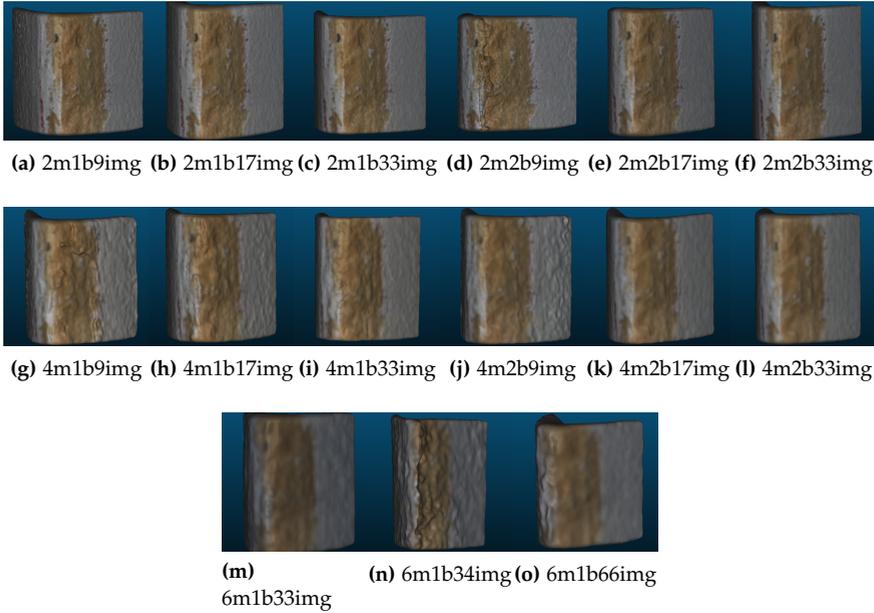


**Fig. D.5:** Examples of the found and matched features of images, from 6 meter datasets that failed to reconstruct correctly. Figure D.5a is from the 9 image dataset, Figure D.5b is from the 17 image dataset, Figure D.5c is from the 9 image two vertical bands dataset and Figure D.5d shows an image from the 33 image dataset that was successfully reconstructed for comparison.

The results from the other 15 datasets are shown in Figure D.6. All reconstruction were scaled to absolute real world scale in mm, before any analysis is conducted on them. From the figure, we can see that some of the results have geometrical surface and shape errors. To properly represent the quality of the reconstructions, the need to be analysed. Because of a lack of high quality ground truth representation of the captured blade, two types of indirect comparison analysis are conducted as part of the paper. The first one focuses on the quality of the overall blade shape, based on the 2D cross-section from

## D.4. Results

the reconstructions. For this we build a mathematical model of a 2D cross-section of the blade, as a ground truth and compare the 2D cross-sections from the reconstructions to it. The second one is directed on the 3D surface roughness features of the blade segments - comparing how both the visual and geometrical representation of the roughness changes and degrades with the different datasets.

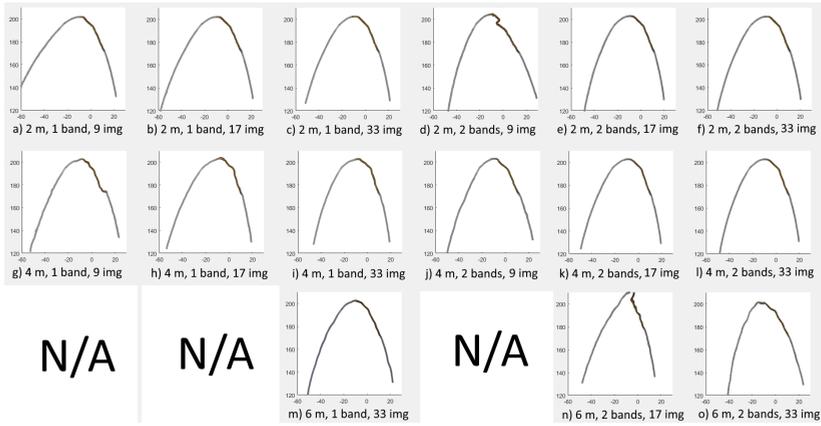


**Fig. D.6:** Reconstruction result mesh for all 15 datasets. The analysis is performed on both the geometrical and visual information present on these meshes. The blades follow a naming convention of - [distance], [number of bands], [number of images].

### D.4.1 2D Shape Analysis

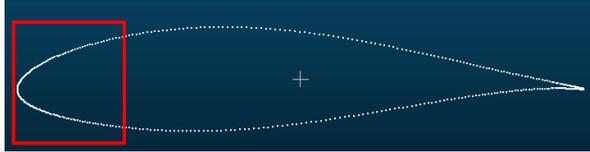
To analyse the accuracy with which the SfM reconstruction has captured the overall shape of the wind turbine blade, without having a high resolution scan of it, a 2D analysis of blade cross-sections is chosen. Each of the 3D reconstructions is oriented such that the Z-axis is pointed upwards. A 2D cross-section in the middle of each blade at the same height is then extracted. All 15 resultant cross-sections are shown in Figure D.7. From the figure, it can be seen that some of the reconstructions have strong shape deformations. This errors will be quantified below.

For testing the accuracy of the captured overall shape, a synthetic 2D



**Fig. D.7:** The resultant 2D cross-sections, extracted from each of the 15 reconstructions. Some of the cross-sections have strong shape deformations, which need to be quantified, by comparison to a ground truth shape. The blades follow a naming convention of - [distance], [number of bands], [number of images].

cross-section is mathematically computed, using the same physical measurements as the reconstructed blade. This is done using the Joukowski airfoil transform method. To do this the chord length, maximum thickness and camber of the captured blade are physically measured. The resultant blade 2D cross-section shape can be seen in Figure D.8. As only the blade edge is reconstructed, as part of the experiments, the trailing edge of the shape is removed.



**Fig. D.8:** The produced blade, made by the Joukowski airfoil transformation method, using the measurements of the real blade. The red square indicates the part of the blade used for comparison with the reconstructions.

To compare the mathematical ground truth to the reconstructed cross-sections, the open source software CloudCompare [41] is used. The ground truth and all reconstructed cross-sections are saved as X, Y point clouds and input to the program. Each of the reconstructions is then finely registered to the ground truth using an iterative closest point algorithm (ICP), which is explicitly restricted to only 2 axes of the blades and rotation around the Z axis (yaw) of the shapes. Scaling is also restricted, as the blades are scaled to real

#### D.4. Results

world sizes, before the ICP algorithm is applied. Once each reconstructed 2D cross-section is registered, mean and standard deviation of the distances between it and the ground truth is calculated. These values are given in Table D.1.

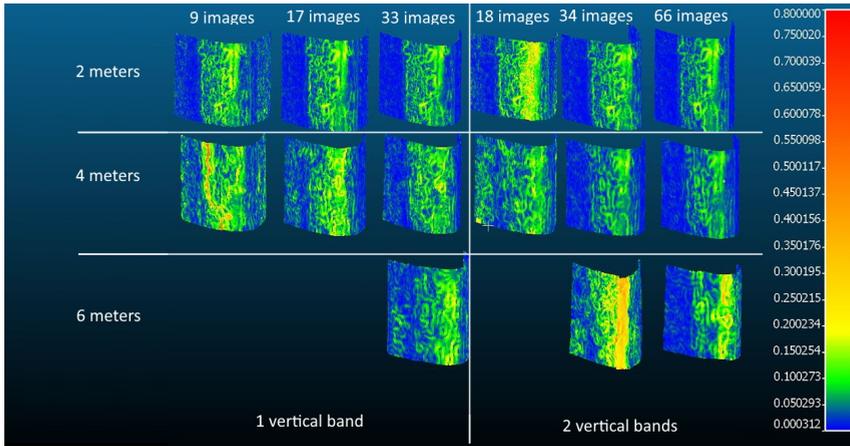
**Table D.1:** Results from the 2D analysis of the shape of the reconstructed blades. The mean and standard deviation of the distances between the ground truth and reconstructed blade cross-sections are in [mm]. The blades follow a naming convention of - [distance], [number of bands], [number of images].

	Mean [mm]	Std.Dev. [mm]
2m1b9img	2.582	2.424
2m1b17img	0.986	0.793
2m1b33img	0.438	0.656
2m2b9img	1.334	0.963
2m2b17img	0.870	0.573
2m2b33img	0.528	0.324
4m1b9img	1.407	1.333
4m1b17img	0.986	0.654
4m1b33img	0.878	0.727
4m2b9img	0.917	0.629
4m2b17img	0.818	0.617
4m2b33img	0.971	0.553
6m1b33img	0.998	0.827
6m2b17img	3.559	1.549
6m2b33img	2.024	1.076

From Table D.1, it can be seen that the best results are achieved, with a high horizontal angular coverage of 33 images per vertical band. When introducing a second band the overall blade shape tends to become less correctly represented with a decrease of 21% at 2m and a decrease of 11% at 4m in the results, but the standard deviation also becomes smaller, giving a more stable overall result. On the other end using only 9 images for either one or two bands tends to perform the worst for capturing the overall shape. An introduction of a second band of images for the 9 image horizontal angular coverage produces a 48% better result at 2m and 35% better at 4m. This demonstrates that, even though capturing the whole surface from many directions is important for a successful reconstruction of the overall shape of an object, introducing too many images, can result in lowering the overall quality, because of the added chance that many of detected features might be noisy or incorrectly matched. It can also be seen that at 6m, the resolution of the camera starts to not be enough for capturing enough details, resulting in worse reconstruction results.

### D.4.2 3D Surface Roughness Analysis

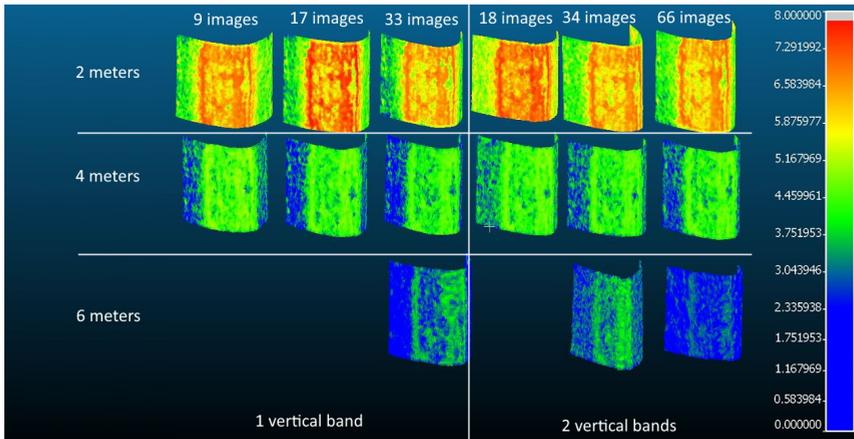
The accuracy, with which each of the reconstructions has captured the roughness of the blade surface also needs to be evaluated. The lack of high resolution ground truth 3D capture of the blade, makes absolute roughness comparison impossible, so for this paper we focus on a comparative analysis between all the SfM reconstructions. This way the possible degradation patterns in the visual and geometrical representation can be detected. By looking at the images taken from the wind turbine blade (Figure D.3), a number of observation can be made and used as the grounds for the analysis:



**Fig. D.9:** The computed DON metric for all 15 reconstructions. All the values are normalized. It can be seen that with the longer distances more geometrical errors become present on the smooth parts of the blade and this process is more severe on the 1 vertical band datasets. Also, a number of blades have large geometrical defects on or close to the edge

1. The geometrical roughness of the 3D reconstruction for damaged regions should be larger and with more variation, because of the multitude of holes and delaminated blade surfaces, as seen by the damaged regions in examples of Figure D.6.
2. The color of the clean areas of the blade should be more uniform than in damaged areas
3. The damaged and clean areas are uniform in their appearance, without any large holes or completely clean, smooth surfaces. If any large geometrical deformities or overly smooth surfaces exist on the reconstruction, they are caused by reconstruction noise and incorrectly captured details, as seen in Figure D.6n and Figure D.6d

## D.4. Results



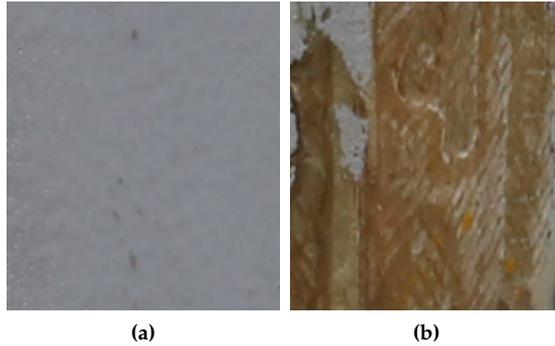
**Fig. D.10:** The computed Entropy metric for all 15 reconstructions. All the values are normalized. The color difference between rough and smooth parts of the blade becomes smaller with larger distances, as the surface is represented by much less pixels in each image. The roughness is also much more represented in the reconstructed surfaces view from two vertical viewpoints.

We look at both the color and the geometrical representation of the wind turbine blade, as SfM has the possibility to output textured meshes and color point clouds, which can be useful for damage detection and roughness evaluation. For each of the measurements the local area around each 3D point needs to be used. A KdTree [42] for faster binary searching is calculated for each of the reconstructions, using the open3D library [43]. For capturing the geometrical surface roughness representation, the difference of normal (DON) algorithm [44] is selected, because of its robustness and straightforwardness. Rough and damaged would give higher uniform DON response, compared to smooth clean surfaces. On the other hand singular peaks of very high DON measurements, can indicate incorrect and noisy reconstruction. For the algorithm, first two radii of different sizes are selected  $r_1$  and  $r_2$ . It is recommended [44] that the difference between the two radii ( $\frac{r_2}{r_1}$ ) should be set to 10, for capturing enough smaller surface details. The difference between the two  $\Delta$  is then defined using Equation D.4 [44], where  $\hat{n}(p, r)$ , is the normal at point  $p$ , using the radius  $r$ .

$$\Delta = \frac{\hat{n}(p_i, r_1) - \hat{n}(p_i, r_2)}{2} \quad (\text{D.4})$$

The normalized results from computing the DON metric for each of the 15 reconstruction can be seen in Figure D.9. The farther the distance becomes between the camera and blade segment, the more noise structures, become apparent on the smooth regions of the blade, while the rough, damaged areas lose their smaller details, which are replaced, by approximated larger struc-

tures. In addition, the introduction of more bands gives more pronounced details to both smooth and rough regions. Some of the reconstructions can be set as outliers - the 18 images at 2m, the 9 images at 4 meters and the 34 images at 6m, as they exhibit large geometrical errors, on their surface. These geometrical errors are much more prevalent at 6 meters, which supports the conclusions taken from the failed reconstructions, that not enough feature points can be detected and matched at that distance, even when a lot of images from different viewpoint are present. Excessive number of images, actually harms the final result, when looking at the 34 and 66 image datasets.



**Fig. D.11:** Selected clear and rough surface patches for calculating the average DON and entropy values, as well as the standard deviation of the values

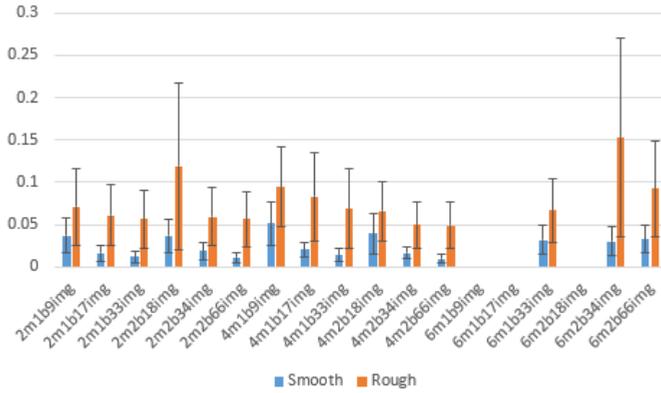
To evaluate, how the captured color detail of the reconstructed surfaces change with each of the datasets, the entropy [45] of the intensity information of each 3D point is also calculated. The higher entropy would show more pronounced changes in the visual color details captured in the reconstruction. This in turn will indicate the presence of more smaller surface details and roughness. It has been shown to give good results for finding surface roughness for denoising [46] and it can be easily calculated for local 3D surface patches. The surface of each reconstruction is separated into a number of areas with a specific radius. The entropy of the color information for each of these areas is calculated using Equation D.5, where  $P_i$  is the probability of a specific intensity occurring at point  $p$  and  $n$  is the number of vertices in the area around each point.

$$H = - \sum_{i=1}^n P_i \log_2 P_i \quad (\text{D.5})$$

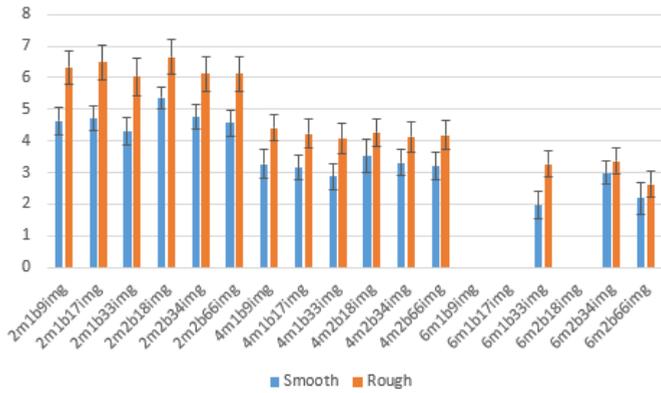
The normalized results from the color entropy calculation for each of the 15 reconstructions are given in Figure D.10. The results show that the color

#### D.4. Results

intensity diversity is drastically lowered at longer distances, making the surface visible details to blend. This is especially noticeable, when comparing the 33 and 66 image dataset results for 2 and 6m. The color information for smaller details on both the smooth and rough regions, is better represented, when using multiple bands compared to just one band, showing that multiple view directions are required, if the reconstruction needs to contain both geometrical and color information for surface inspection.



(a) DON



(b) Entropy

**Fig. D.12:** The average and standard deviation of the DON and entropy values calculated from the selected smooth and rough patches, seen in Figure D.11

To illustrate the changes in the surface and visual roughness for each of the reconstructions, two patches are selected from the blade - one representing a clean area of low roughness and one representing a damaged area of high roughness (Figure D.11). The average and standard deviation of the

DON and color entropy values for these patches are calculated and shown in Figure D.12. We can see that in the case of smooth patch, the average DON measurement follows a trend of lowering its values with the number of images in the horizontal direction. The distance between the camera and blade does not drastically change the calculated geometrical roughness. For the rough areas the same trends can be seen, but less pronounced. The overall DON metric also increases, indicating that the captured structures become larger and deviate more from the real object's roughness. The standard deviation of the rough regions, is also quite larger than the smooth ones, for the most part indicating that the roughness details are captured. The reconstructions with large geometrical errors, are the outlier one exhibiting very high DON values with high standard deviation. For the entropy calculation a clear trend of lowering the average values is also observed, for both the rough and smooth regions, with the smooth ones exhibiting a less steep decline. This can be explained by the loss of the smaller, finer detail associated with the damages on the surface, as the distance become bigger. The standard deviation of the entropy calculation is also lowered meaning that the variation in the colors, indicating surface roughness also is lowered.

## D.5 Conclusion and Future Work

Blade inspection is a vital part of the wind energy industry and detecting damages or an increased roughness profile on wind turbine blades, is instrumental for keeping energy production high. The use of 3D point cloud data for inspection is becoming more wide spread, as it gives another dimension to the captured information and can be used to more easily detect small changes in blade roughness.

In this paper we carry out a benchmark on the use of SfM reconstructions for performing such inspections. A decommissioned wind turbine blade segment is selected and additional roughness and damaged areas are introduced to its surface by sandblasting it. We created 18 datasets, by varying the distance to the blade, the amount of horizontal images and the amount of vertical bands of images. All images were taken in an outdoor environment, from a vertically oriented blade, to ensure a capturing setup as close as possible to real life.

We conducted both a 2D and 3D surface analysis on the resultant reconstructions. The 2D analysis is done on slices extracted from each blade segment and compared to a mathematically computed ground truth blade slice. Through this analysis we demonstrated that the more horizontal overlap there is between the images, the better the shape of the blade is represented. While better vertical angular coverage ensuring that all sides of the blade are represented with a proper shape, which is especially important

when not enough horizontal coverage exists. The distance from the blade lowers the results, with larger problems observed at 6m with the current capturing setup.

The 3D analysis is aimed at analysing how the roughness representation is changed, with the different capturing setups. Because of lack of a high resolution ground truth scan of the blade surface, we choose to focus on analysing how the geometrical and visual representation of the blade surfaces in the different reconstructions changes. For this we choose two metrics - the difference of normals for representing the geometrical variation of the surface and the color intensity entropy for representing the visual variance, which can be useful for detecting damages and roughness changes. Through analysis of smooth and rough patches of each of the reconstruction we demonstrate that the geometrical and color representation of smooth areas follows a strict trend for lowering the perceived roughness with the amount of horizontal image overlap and with increasing the distance between the blade and camera. We can see that depending on the chosen of the capturing setup factors, SfM reconstruction can go from graceful degradation of resolution and captured features, to a tipping point, at which the algorithm cannot capture enough data from the environment and the surface, which results in complete failure to produce an output. This "tipping point" is highly dependent on the used camera, captured surface properties, like texture, shape, size, material, as well as the lighting conditions. In the case of the current paper's experiments, this tipping point can be observed at going from 4 to 6m distance. This can be alleviated to some extent by introducing more images, with more diverse horizontal and vertical angular coverage, but even then when the reconstruction produces a result, there is large possibility that the captured data is too noisy for capturing the surface characteristics.

Finally we show that if both the overall shape and smaller roughness details need to be captured from a wind turbine blade, the best capturing setup would require between 2 and 4 meters of distance from the blade, with at least 17 images captured in a semi-circle. If less images are taken horizontally, then it is required that more vertical bands need to be taken to capture enough information of the surface. Finally, capturing too many images can be detrimental to the results, as it can introduce too many noise features and outliers, which can skew the reconstructed surface and calculated camera positions.

For future work we aim to gather high resolution scans of different types of blades and use them as ground truth for a more in-depth analysis, using the proposed metrics. In addition, we will want to introduce additional capturing setup differences in the form of varying illumination, camera focal lengths and settings.

## References

- [1] K. Yang, "Geometry design optimization of a wind turbine blade considering effects on aerodynamic performance by linearization," *Energies*, vol. 13, no. 9, p. 2320, 2020.
- [2] F. Song, Y. Ni, and Z. Tan, "Optimization design, modeling and dynamic analysis for composite wind turbine blade," *Procedia Engineering*, vol. 16, pp. 369–375, 2011.
- [3] W. Han, J. Kim, and B. Kim, "Effects of contamination and erosion at the leading edge of blade tip airfoils on the annual energy production of wind turbines," *Renewable energy*, vol. 115, pp. 817–823, 2018.
- [4] M. Ge, H. Zhang, Y. Wu, and Y. Li, "Effects of leading edge defects on aerodynamic performance of the s809 airfoil," *Energy conversion and management*, vol. 195, pp. 466–479, 2019.
- [5] M. Darbandi, A. Mohajer, A. Behrouzifar, R. Jalali, and G. E. Schneider, "Evaluating the effect of blade surface roughness in megawatt wind turbine performance using analytical and numerical approaches." International Conference on Heat Transfer, Fluid Mechanics and Thermodynamics, 2014.
- [6] L. Bartolomé and J. Teuwen, "Prospective challenges in the experimentation of the rain erosion on the leading edge of wind turbine blades," *Wind Energy*, vol. 22, no. 1, pp. 140–151, 2019.
- [7] R. Raišutis, E. Jasiūnienė, R. Šlīteris, and A. Vladišauskas, "The review of non-destructive testing techniques suitable for inspection of the wind turbine blades," *Ultragarsas" Ultrasound"*, vol. 63, no. 2, pp. 26–30, 2008.
- [8] M. Doroshtnasir, T. Worzewski, R. Krankenhagen, and M. Röllig, "On-site inspection of potential defects in wind turbine rotor blades with thermography," *Wind Energy*, vol. 19, no. 8, pp. 1407–1422, 2016.
- [9] A. Reddy, V. Indragandhi, L. Ravi, and V. Subramaniaswamy, "Detection of cracks and damage in wind turbine blades using artificial intelligence-based image analytics," *Measurement*, vol. 147, p. 106823, 2019.
- [10] S. Gerbino, D. M. Del Giudice, G. Staiano, A. Lanzotti, and M. Martorelli, "On the influence of scanning factors on the laser scanner-based 3d inspection process," *The International Journal of Advanced Manufacturing Technology*, vol. 84, no. 9-12, pp. 1787–1799, 2016.

## References

- [11] A. H. Ahmadabadian, S. Robson, J. Boehm, M. Shortis, K. Wenzel, and D. Fritsch, "A comparison of dense matching algorithms for scaled surface reconstruction using stereo camera rigs," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 78, pp. 157–167, 2013.
- [12] J. Xu, N. Xi, C. Zhang, Q. Shi, and J. Gregory, "Real-time 3d shape inspection system of automotive parts based on structured light pattern," *Optics & Laser Technology*, vol. 43, no. 1, pp. 1–8, 2011.
- [13] Y. He, B. Liang, Y. Zou, J. He, and J. Yang, "Depth errors analysis and correction for time-of-flight (tof) cameras," *Sensors*, vol. 17, no. 1, p. 92, 2017.
- [14] M. S. Nielsen, I. Nikolov, E. K. Kruse, J. Garnæs, and C. B. Madsen, "High-resolution structure-from-motion for quantitative measurement of leading-edge roughness," *Energies*, vol. 13, no. 15, p. 3916, 2020.
- [15] S. Stent, R. Gherardi, B. Stenger, K. Soga, and R. Cipolla, "Visual change detection on tunnel linings," *Machine Vision and Applications*, vol. 27, no. 3, pp. 319–330, 2016.
- [16] J. E. N. Masson and M. R. Petry, "Comparison of mesh generation algorithms for railroad reconstruction," in *Autonomous Robot Systems and Competitions (ICARSC), 2017 IEEE International Conference on*. IEEE, 2017, pp. 266–271.
- [17] A. A. Alhasan, K. Younkin, and D. J. White, "Comparison of roadway roughness derived from lidar and sfm 3d point clouds," 2015.
- [18] T. Dewez, D. Girardeau-Montaut, C. Allanic, and J. Rohmer, "Facets: A cloudcompare plugin to extract geological planes from unstructured 3d point clouds," 2016.
- [19] D. Fawcett, B. Azlan, T. C. Hill, L. K. Kho, J. Bennie, and K. Anderson, "Unmanned aerial vehicle (uav) derived structure-from-motion photogrammetry point clouds for oil palm (*elaeis guineensis*) canopy segmentation and height estimation," *International Journal of Remote Sensing*, vol. 40, no. 19, pp. 7538–7560, 2019.
- [20] G. Kyriakaki, A. Doulamis, N. Doulamis, M. Ioannides, K. Makantasis, E. Protopapadakis, A. Hadjiprocopis, K. Wenzel, D. Fritsch, M. Klein *et al.*, "4d reconstruction of tangible cultural heritage objects from web-retrieved images," *International Journal of Heritage in the Digital Era*, vol. 3, no. 2, pp. 431–451, 2014.
- [21] A. Corradetti, K. McCaffrey, N. De Paola, and S. Tavani, "Evaluating roughness scaling properties of natural active fault surfaces by means of multi-view photogrammetry," *Tectonophysics*, vol. 717, pp. 599–606, 2017.

## References

- [22] D. Zhang, R. Watson, G. Dobie, C. MacLeod, A. Khan, and G. Pierce, "Quantifying impacts on remote photogrammetric inspection using unmanned aerial vehicles," *Engineering Structures*, p. 109940, 2020.
- [23] I. Reljić, I. Dundjer, and S. Seljan, "Photogrammetric 3d scanning of physical objects: Tools and workflow," *TEM Journal*, vol. 8, no. 2, p. 383, 2019.
- [24] G. Gabara and P. Sawicki, "Multi-variant accuracy evaluation of uav imaging surveys: A case study on investment area," *Sensors*, vol. 19, no. 23, p. 5229, 2019.
- [25] T. P. Kersten, D. Omelanowsky, and M. Lindstaedt, "Investigations of low-cost systems for 3d reconstruction of small objects," in *Euro-Mediterranean Conference*. Springer, 2016, pp. 521–532.
- [26] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, p. 78, 2017.
- [27] J. Schöning and G. Heidemann, "Taxonomy of 3d sensors," *Argos*, vol. 3, no. P100, pp. 9–10, 2016.
- [28] I. Nikolov and C. Madsen, "Benchmarking close-range structure from motion 3d reconstruction software under varying capturing conditions," in *Euro-Mediterranean Conference*. Springer, 2016, pp. 15–26.
- [29] M. Marčiš, "Quality of 3d models generated by sfm technology," *Slovak Journal of Civil Engineering*, vol. 21, no. 4, pp. 13–24, 2013.
- [30] N. D'Amico and T. Yu, "Accuracy analysis of point cloud modeling for evaluating concrete specimens," in *SPIE Smart Structures and Materials+ Nondestructive Evaluation and Health Monitoring*. International Society for Optics and Photonics, 2017, pp. 101 691D–101 691D.
- [31] M. Favalli, A. Fornaciai, I. Isola, S. Tarquini, and L. Nannipieri, "Multiview 3D reconstruction in geosciences," *Computers & Geosciences*, vol. 44, pp. 168–176, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0098300411003128>
- [32] M. Smith, J. Carrivick, and D. Quincey, "Structure from motion photogrammetry in physical geography," *Progress in Physical Geography*, vol. 40, no. 2, pp. 247–275, 2016.
- [33] S. Bianco, G. Ciocca, and D. Marelli, "Evaluating the performance of structure from motion pipelines," *Journal of Imaging*, vol. 4, no. 8, p. 98, 2018.

## References

- [34] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [35] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.
- [36] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [37] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *International workshop on vision algorithms*. Springer, 1999, pp. 298–372.
- [38] C. Bak, M. Gaunaa, A. S. Olsen, and E. K. Kruse, "What is the critical height of leading edge roughness for aerodynamics," in *Journal of Physics: Conference Series*, vol. 753, no. 2, 2016, p. 022023.
- [39] J. P. Dandois, M. Baker, M. Olano, G. G. Parker, and E. C. Ellis, "What is the point? evaluating the structure, color, and semantic traits of computer vision point clouds of vegetation," *Remote Sensing*, vol. 9, no. 4, p. 355, 2017.
- [40] Agisoft, "Metashape (photoscan)," <http://www.agisoft.com/>, 2010, accessed: 2020-06-15.
- [41] D. Girardeau-Montaut, "Cloudcompare-open source project," *Open-Source Project*, 2011.
- [42] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.
- [43] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.
- [44] Y. Ioannou, B. Taati, R. Harrap, and M. Greenspan, "Difference of normals as a multi-scale operator in unorganized point clouds," in *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*. IEEE, 2012, pp. 501–508.
- [45] A. Shiozaki, "Edge extraction using entropy operator," *Computer Vision, Graphics, and Image Processing*, vol. 36, no. 1, pp. 1–9, 1986.
- [46] J. Navarrete, D. Viejo, and M. Cazorla, "Color smoothing for rgb-d data using entropy information," *Applied Soft Computing*, vol. 46, pp. 361–380, 2016.

## References

# Paper E

## Performance Characterization of Absolute Scale Computation for 3D Structure from Motion Reconstruction

Ivan Nikolov and Claus Madsen

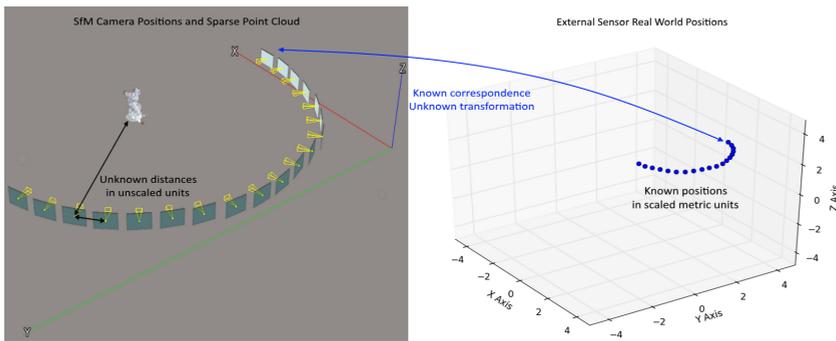
The paper has been published in the  
*Proceedings of 14th International Joint Conference on Computer Vision, Imaging  
and Computer Graphics Theory and Applications*, pp. 884-891, 2019.

© 2019 SCITEPRESS Digital Library  
*The layout has been revised.*

## Abstract

*Structure from Motion (SfM) 3D reconstruction of objects and environments has become a go-to process, when detailed digitization and visualization is needed in the energy and production industry, medicine, culture and media. A successful reconstruction must properly capture the 3D information and it must scale everything to the correct scale. SfM has an inherent ambivalence to the scale of the scanned objects, so additional data is needed. In this paper we propose a lightweight solution for computation of absolute scale of 3D reconstructions by using only a real-time kinematic (RTK) GPS, in comparison to other custom solutions, which require multiple sensor fusion. Additionally, our solution estimates the noise sensitivity of the calculated scale, depending on the precision of the positioning sensor. We first test our solution with synthetic data to find how the results depend on changes to the capturing setup. We then test our pipeline using real world data, against the built-in solutions used in two state-of-the-art reconstruction software. We show that our solution gives better results, than both state-of-the-art solutions.*

## E.1 Introduction



**Fig. E.1:** The two sets of corresponding points. On the left an output screen from 3D reconstruction program with the camera positions and a sparse point cloud, with unknown scale. On the right the same camera positions in a scaled real world metric units from an external sensor. Establishing the absolute scale of the reconstruction involves estimation the transformation, which will transfer the left set of points to the right.

### E.1.1 Object 3D Reconstruction

With the emergence of more and more powerful CPU and GPUs, SfM software solutions have become widespread and easier to use. This gives both more specialized industry, medicine and culture preservation users the possibility to quickly capture objects and environments. Due to the nature of

SfM, to create a detailed reconstruction of both object and texture, users need only a camera and the software. This gives SfM solutions the edge, compared to other low-cost 3D reconstruction solutions, like the ones based on time-of-flight [1], structured light [2] or stereo cameras [3]. These solutions require appropriate hardware, together with the specialized software, which gives them a larger overhead, for users to get into. Examples of 3D reconstruction using these methods are extensively benchmarked by [4], [5].

### **E.1.2 State of the Art**

An important requirement for the state of the art SfM software is for it to be both versatile and robust. This is especially true for images taken in environments with varying conditions and containing objects with different shapes and sizes. Many of the state of the art SfM solutions fall in the category of open-source software like OpenSfM [6], COLMAP [7], etc. Other SfM solutions are developed as part of commercial products like ContextCapture [8], PhotoScan(Metashape) [9] and RealityCapture [10]. All of these solutions contain a whole processing pipeline going from the input images to a dense point cloud and mesh. One drawback that SfM has, is the ambiguity of the scale of the reconstructed object. The 2D information extracted from images, does not allow determining of the absolute scale of the scanned object. For obtaining this essential information, additional information is needed from the user or from external sensors.

This is why the programs also contain different built-in solutions for scaling the final model. In most cases these solutions are either using markers or manual distance measurement. This works only when there is access to the reconstructed objects or surfaces. This means that they are unfeasible for scanning structures with drones or scanning hard to reach or dangerous places. Another method is using the GPS positions for finding the absolute scale of the object, but this way does not characterize the performance of the scaling and do not take into consideration external factors, which can influence the precision of the scaling.

### **E.1.3 Using External Sensor Data for Determining Scale and Noise Sensitivity**

We propose a solution for determining the absolute scale for 3D SfM reconstructions using GPS positioning information, enhanced by RTK for a more precise estimation of the camera for each image taken. Others have proposed method for using GPS and RTK [11], [12] for georeferencing and enhancing the SfM reconstruction workflow, but they do not focus on the scale of the reconstructed objects. Our method is aimed at being used as part of a unmanned aerial vehicle (UAV) solution for scanning and 3D reconstruction

of hard to reach surfaces and objects. It also works only with GPS data, in contrast with other methods using sensor fusion [13]. As the method is aimed for industrial and historical preservation use, not only is the absolute scale needed, but also calculating the uncertainty of said scale, as well as determining how capturing conditions and external factors might influence it. As an example, we show that when you have 3 captured horizontally spaced camera positions from the reconstruction process, a scaled distance of  $100mm$  on a reconstructed object can be with uncertainty of  $0.1mm$ . While the same scaled distance, when calculated from 18 captured horizontally spaced camera positions has uncertainty of  $0.007mm$ .

Our *main* contribution is this combination of estimating the absolute scale and its sensitivity to noise, which in the end gives both precise and robust results.

To test our approach, we first analyze the sensor and determine its accuracy and precision. We then do a series of simulated test scenarios to get a baseline of the expected performance. Finally, we test the method in a real world testing scenario and compare results to the scaling results produced by the two 3D reconstruction programs - ContextCapture and PhotoScan(Metashape). We demonstrate that our method gives better results than the state-of-the-art, while also providing a reliable uncertainty metric.

## E.2 Methodology

### E.2.1 SfM Pipeline

To understand the proposed workflow, the SfM pipeline needs to be first explored. SfM relies on information captured from multiple images around the scanned objects. Features are extracted from each image and matched. Normally algorithms like SIFT [14], SURF [15] are used. From these matched points a sparse 3D point cloud can be triangulated using bundle adjustment [16] and the camera positions can be back-projected. From these a dense point cloud and subsequent mesh can be created. The problem is that there is no information in 2D images alone on how big the scanned object is - is it a city or a model of a city? This is also reflected in the calculated camera positions.

### E.2.2 Least-Squares Transformation Estimation

When capturing the images, the positions of the cameras can be saved in real world coordinates. The GPS-RTK can be directly positioned on the camera or on a drone carrying the camera. To calculate the real world scale of the reconstruction, the transformation between the two sets of coordinates needs to be determined - the ones calculated by the SfM software and the ones

given by the GPS-RTK. This is shown in Figure E.1, where an output of SfM software is shown on the left side and the GPS-RTK points are shown on the right side.

Because there is clear a correspondence between the SfM camera positions and the GPS-RTK positions and the unknown transformation consists only of translation, rotation and uniform scaling, a simple least-squares estimation algorithm is considered. An implementation of the classical algorithm by [17] is chosen and customized for the needs of the paper. For the algorithm to work the two point sets need to have non-collinear points and no outliers.

We need to also take into consideration the problem of the lever-arm offset between the GPS antenna and the camera [18]. As an initial calibration step the real-world distance between the two is measured and used as an additional input for the Least-Squares estimation algorithm.

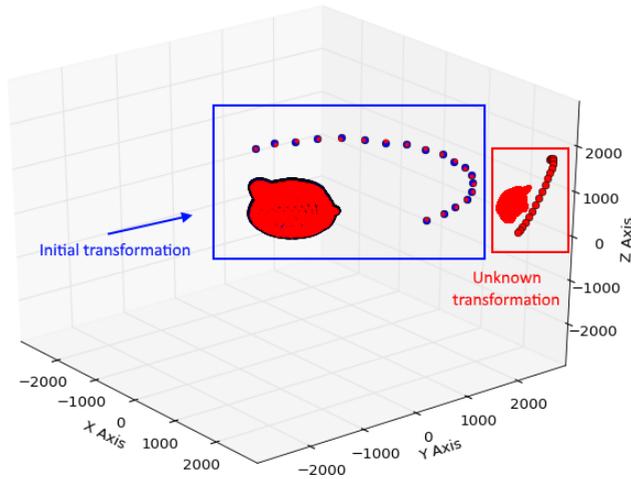
$$a_i = T(b_i), \quad a_i \in A, \quad b_i \in B \quad (\text{E.1})$$

$$T = \begin{bmatrix} sR_{11} & sR_{12} & sR_{13} & t_1 + x_{off} \\ sR_{21} & sR_{22} & sR_{23} & t_2 + y_{off} \\ sR_{31} & sR_{32} & sR_{33} & t_3 + z_{off} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{E.2})$$

If the two point sets are  $A = [a_1, a_2, \dots, a_m]$ , for the known one and  $B = [b_1, b_2, \dots, b_m]$  for the unknown one, where each set is comprised of  $m$  number of points and each point has a  $x, y, z$  components. Then the transformation matrix  $T$  between the two needs to be calculated, such that it satisfies Equation E.1. To do that, the sum of squared errors  $e^2$  shown in Equation E.3 from [17], needs to be minimized, where  $s$  is the scale,  $R$  is the rotation and  $t$  is the translation component of the transformation matrix (Equation E.2). The real world offset between the two sensors is given as  $x_{off}, y_{off}, z_{off}$  inputs.

$$e^2(R, t, s) = \frac{1}{m} \sum_{i=1}^m \|a_i - (sRb_i + t)\|^2 \quad (\text{E.3})$$

To test out the algorithm's results, a synthetic point set of 18 points is created. The number of points is chosen such that it coincides with the tests, performed later. A new set of points is then created, by giving the point set, a random translation, rotation and scaling. The two sets are used in the least-square estimation algorithm. The result estimated transformation matrix is exactly the same as the one introduced to the ground points to create the unknown ones. This is seen in Figure E.2, where the estimated transformation matrix is used on the Utah teapot, to transform it to the coordinate's initial position, together with the unknown positions.



**Fig. E.2:** Visualization of the output of the least-squares transformation estimation algorithm. The initial position, orientation and scale are first transformed to "unknown" ones. The estimation algorithm is then used to find the transformation from the "unknown" one to the initial one. The Utah teapot is added for easier visualization

In the real world this is not the case, as measuring equipment is always a subject to additive noise. This needs to be taken into consideration, when using the least-squares estimation algorithm. This will transform Equation E.3, where  $C = A + N$ , with  $A$  being the known locations and  $N = [n_1, n_2, \dots, n_m]$  is the added noise component, with each noise  $n_i = [n_x, n_y, n_z]^T$ . The next subsections will verify the sensor readings and model the noise.

### E.2.3 Verifying Sensor Readings

For the paper the sensor provided by DJI [19] is used, as it has a very small margin of uncertainty in the positioning information - less than 0.02m in horizontal direction and 0.03m vertically, in good weather conditions. This precision needs to be verified, before using it. Because the sensor works only when attached to a drone controller, the whole platform is used for the test. The sensor is started and its readings are saved each second for a period of 5 minutes. The readings are taken when the whole platform is on the ground, to eliminate inconsistencies from the readings when the platform is in motion. The sensor is then manually moved to another location and the readings are again taken. The calculated position standard deviation for the first point is 0.0175m in horizontal direction and 0.0244m in height and for the second point the standard deviations are 0.0174m and 0.0251m respectively. The values are thus in the interval given in the documentation by DJI. With

the real world positioning uncertainty verified, the next step is to create a number of synthetic testing scenarios, where the uncertainty is used as a noise component. These scenarios are used to investigate aspects of how the GPS-RTK noise influences scale noise.

## E.2.4 Synthetic Testing Scenarios

For the synthetic tests to be as close to a real world test, the point sets are setup as real SfM capturing positions. The tests are designed to determine the amount of camera positions needed and the amount of vertical camera bands. To gather enough variation in the calculated scale after the noise input in each of the tests, the least-squares estimation step is done a number of times, each time with a different sampling of noise input.

### Number and sampling of image positions

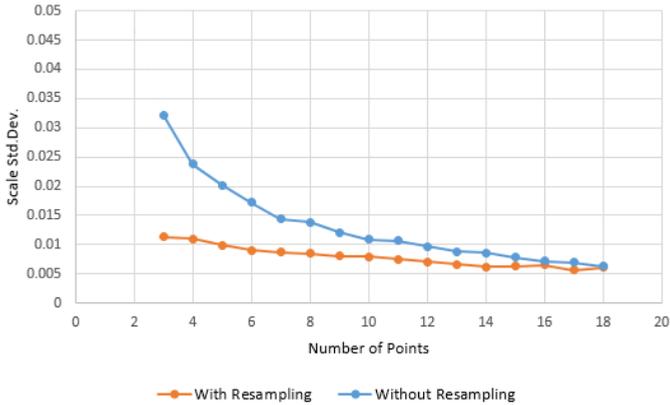
The first synthetic test scenario looks at how the number of input camera positions affects the results of the scale factor. The least-squares estimation method requires at least two positions for estimation of the transformation. In the paper by [20], a circular pattern of images is used, with the position of each image, changing by 20 degrees. This gave 18 images per circular pattern. For a simpler and easier image capture for the real world test scenarios described in the later sections, the circular pattern is changed to a semi-circular one, leaving the number of image positions to 18 again, giving a 10 degree separation between them. This gives the final testing interval - 3 to 18 positions. The minimum number of positions is set to 3, as at least 3 points are needed to estimate the 3D transformation. To test this we start with the full number of 18 positions going from 0 to 180 degrees. Then every time we lower the number of positions we do not just remove the last one, but we resample the left ones so they always cover the whole interval of 0 to 180 degrees, but have larger distance between them.

The synthetic test is done once without resampling, starting with 18 positions and removing positions, until only the first two are left. The second run of the test is done with removing and resampling the positions until only the 0 and 180 degree ones are left.

To model the positioning noise for each instance, a random sampling of the uncertainty values captured directly from the GPS-RTK. This of course introduces the problem that not enough data has been captured for a more diverse modeling of the uncertainty. We will address this in the next subsection.

The obtained results are shown in Figure E.3. When resampling the positions, as points are removed the additional separation between points helps with lowering the calculated scale's error. This is especially evident up until

## E.2. Methodology



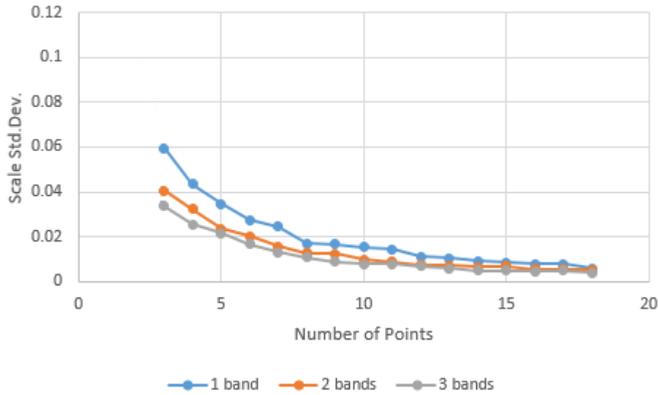
**Fig. E.3:** Results from the resampling versus no resampling synthetic test. Resampling the captured positions so the first and last one are always at 0 and 180 degrees, after removing positioning information drastically lowers the standard deviation of the calculated scale

10 image positions. After that the two methods have comparable result standard deviations, which converge at 16 image positions. This shows that if not all image positions can be captured, it is better that the captured ones have maximum separation. In addition, the standard deviation settles at around 12 or 13 image positions.

### Number of vertical bands

The second test scenario is designed to check how many vertical bands of images and image positions are needed. The previous test showed that worse scale uncertainty is achieved when no resampling of the points is present. This test will explore if better scale uncertainty can be achieved with more vertical separation between the positions, if no resampling is used. The work of [20], shows that three bands of photos give the best possible reconstruction results. The paper however manually scales the output meshes, so no conclusions are given on how the scaling is affected by the bands. To test this we choose to test with one, two and three position bands respectively. This will determine if the additional spatial change between the positions in different bands, given to the least-squares estimator, will make a difference to the calculated scale factor.

The synthetic test is started with one band of vertical separation and 18 camera positions. The number of positions is reduced by one for each test until only 3 positions are left. The same is done for two and three vertical band tests. The estimated scale factor is calculated from each and standard deviation is calculated from all the possible results. The results are given in Figure E.4.



**Fig. E.4:** Results from capturing of positioning data from different number of vertical bands. More vertical bands help with the uncertainty of the scale. Both the larger number of points and the additional spatial information help with that.

**Table E.1:** Change in the scale standard deviation when going from 1 to 2 bands and from 2 to 3 bands for the minimum and maximum number of tested point positions. The change from 2 to 3 bands is almost twice as small showing that the gained accuracy, is not enough to offset the larger amount of data, longer capturing time, etc.

Points	Difference	
	1 and 2 bands (%)	2 and 3 bands (%)
2	43.75	14.60
18	11.26	27.51

As expected, the high standard deviation decreases as we introduce additional vertical positions in the form of more bands. This is both because of the larger number of points and additional vertical separation. If we look at the difference between the standard deviations of the calculated average scale we can see a relation between the number of bands and number of points. The data is given in Table E.1. When more points are present in each band the gains won by going from one to two bands are not big, but if multiple bands need to be taken, then it will be much better to capture three. When less points are present in each band it is necessary to have as much bands as possible, so the benefits from the additional number of points and separation can be felt. To strike a balance between number of bands captured and scale precision gains, we choose to use two bands for the real world testing scenario for testing against the state of the art.

### E.2.5 Covariance Propagating of Positioning Noise

The way the noise is introduced in these synthetic tests and the performance characterization of the scale calculation is found, can be cumbersome, as the test needs to be done a large number of times. A better solution to this is using covariance propagation [21] of the noise. This will give the relation between the uncertainty in each GPS-RTK position and the uncertainty in the final calculated scale. The idea has been shown to give good results [22], as long as there are independent input parameters, which are used in a function - no matter analytically or iteratively found, to calculate a set of output parameters. As each captured GPS-RTK position is used in the calculation of the scale factor through the least-square estimation, this means that we can express the transformation calculation as represented as  $s = f(C)$ , where  $s$  is the estimated scale and  $C$  is the GPS-RTK positioning set together with the introduced noise. We do not use the second positioning set  $B$  obtained from the SfM reconstruction, as an input parameter, as it is treated as a constant. We use the method demonstrated in [22], for determining the covariance matrix of the input parameters. This of course need to be done for each of the three dimensions for each of the points. The standard deviation of the calculated scale will depend both on the standard deviation of the uncertainty of the measured GPS-RTK positions and the first derivative of the function. To find the standard deviation of the scale, the first order approximation needs to be done to the covariance matrix, as seen in Equation E.4 and then used together with the dependence of the scale to the positions in all 3 dimensions, as given in Equation E.5. Where  $\Delta$  is the covariance matrix of  $Q$  and is described as Equation E.6, for each of its dimensions. Thus  $Q$  is a combined vector containing all the dimensional data for each position  $Q = [x_1, y_1, z_1, \dots, x_m, y_m, z_m]^{1 \times 3m}$

$$\Delta = \begin{bmatrix} \sigma_{x_1}^2 & 0 & 0 & \dots & 0 \\ 0 & \sigma_{y_1}^2 & 0 & \dots & 0 \\ 0 & 0 & \sigma_{z_1}^2 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \sigma_{x_m}^2 & 0 & 0 \\ 0 & \dots & 0 & \sigma_{y_m}^2 & 0 \\ 0 & \dots & 0 & 0 & \sigma_{z_m}^2 \end{bmatrix}_{3m \times 3m} \quad (\text{E.4})$$

$$\sigma_s^2 = \frac{\partial s}{\partial Q} \Delta \frac{\partial s^T}{\partial Q} \quad (\text{E.5})$$

$$\frac{\partial s}{\partial Q} = \left[ \frac{\partial s}{\partial x_1} \frac{\partial s}{\partial y_1} \frac{\partial s}{\partial z_1} \dots \frac{\partial s}{\partial x_i} \frac{\partial s}{\partial y_m} \frac{\partial s}{\partial z_m} \right] \quad (\text{E.6})$$

To test if the iterative approach and the covariance propagation approach will yield the same results. Again the testing scenario of subsection E.2.4 is

used, both with and without resampled positioning data. The results can be seen in Figure E.5. The two approaches achieve closely matched results for both positioning data types. The average difference between the calculated scale standard deviation from 3 to 18 points between the two approaches, without resampling positioning data is 12.07%, while with resampling the difference falls to 4.88%. In addition the covariance propagation approach follows a smoother overall curve of progression, meaning less chance for random noise in the estimated scale standard deviation. This also demonstrates that the covariance propagation method gives very accurate estimation of the standard deviation of the scale, while bypassing the assumptions about the nature of the uncertainty's distribution.

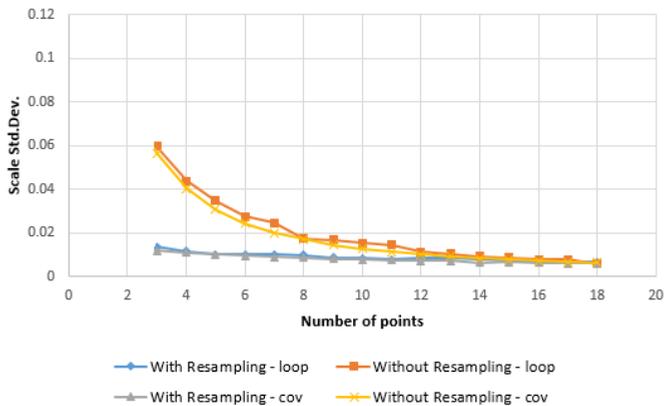


Fig. E.5: Comparing the iterations approach to the covariance propagation approach for calculating scale uncertainty from position uncertainty. The comparison is made for different number of input positions and a different way of resampling the positions

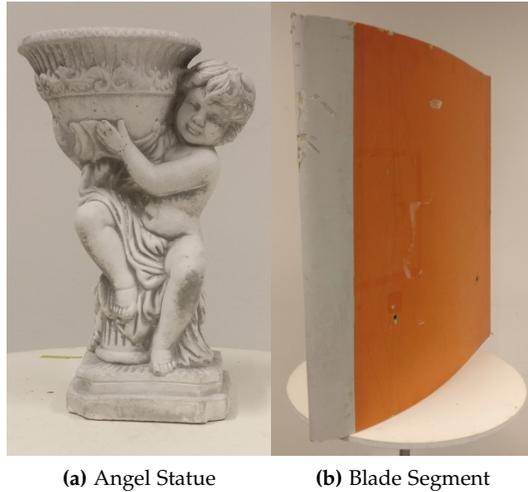
### E.3 Real World Testing

Two objects are chosen for the test. They can be seen in Figure E.6. The objects are chosen because they represent two different 3D reconstruction use cases - the statue represents a digital heritage use case, while the wind turbine blade represents an industrial use case. Both cases require precise scale estimation.

For each of the two objects, two vertical bands of 18 images are taken in a semi-circle. The horizontal separation between the images is 10 degrees, while the vertical separation between bands is 20 degrees. The images are taken with a Canon 5Ds at maximum resolution 8688x5792. This camera is chosen, so enough information is captured from the objects and the chance of the 3D reconstruction failing or having errors is minimized.

The camera positions are manually determined with a laser range finder.

### E.3. Real World Testing



**Fig. E.6:** Two test objects used for 3D reconstruction. Each represent different reconstruction challenges and reconstruction cases

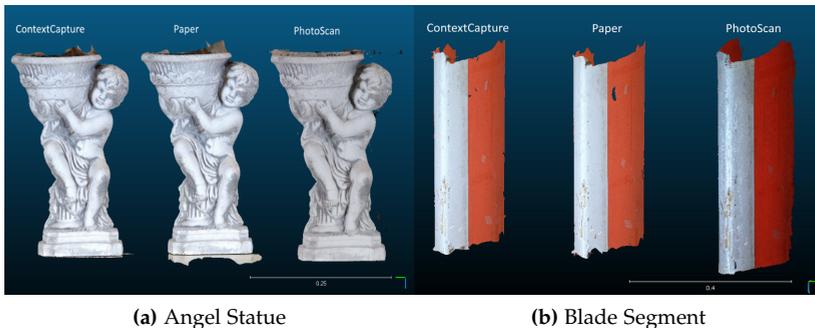
This is done so that any possible random positioning accuracy fluctuations on the GPS-RTK, caused by weather conditions, pressure changes or environment effects are removed. A second reason for this is that this way the experiment can be done in an indoor environment, removing the possible illumination changes that can affect the final reconstruction.

For the reconstruction both PhotoScan(Metashape) and ContextCapture software is used. The two solutions have a number of built-in ways to scale a model - using point markers that the user directly adds to the model and have been measured beforehand, printing marker trackers and putting them around the scanned object and detecting them in the images, adding coordinates to the camera positions from GPS. For testing our proposed solution, we have chosen the method that is most relevant - adding camera positions, together with the images and using them to scale the object.

Because the built-in solutions do not have a measurement of the uncertainty of the scaling, the comparison will be done only on the basis of the calculated scaling factors. For comparing the calculated scale factors, the reconstructed objects will be scaled using these factors and the distance will be measured manually on the real world object and the scaled reconstruction.

As there are no ground truth scaled model to compare the scales from the three methods, a manual measuring of the objects is chosen. A number of parts of the two real world objects are measured with a caliper, which has a resolution of  $\pm 0.02mm$ , when measuring objects below  $100mm$ . The reconstructed and scaled model are imported into CloudCompare [23] for

measuring the same parts. By measuring multiple parts of the models and averaging the difference between the real world measurement and the scaled model measurement, the effects of the possible human errors, while manually measuring are minimized.



**Fig. E.7:** Reconstructed and scaled model. The ContextCapture and PhotoScan(Metashape) reconstructions are scaled using the built-in solutions in the software. For our proposed solution(Paper), the unscaled ContextCapture reconstruction is used as basis. The brightness difference in the models is due to the different ways the programs normalize the texture color. In addition the model from PhotoScan(Metashape) reconstructed a larger portion of the blade and looks larger even though the scale is comparable.

The obtained scaled models are given in Figure E.7. Just by looking at the models, no observable difference can be seen. Table E.2 contains the average measured distance errors from measuring ten different parts in the real world and on the reconstructed objects, as well as the standard deviation from the measurements. The results show that our proposed solution gives better results, because the mean error distance is the lowest compared to the other. Even if we factor in the manual repeated measurement error, shown as the standard deviation of the distance in the table, the results obtained by our method are better or the same as the build in solutions.

**Table E.2:** Average distance error between measurements from the real world object and the reconstructed model, for the two tested objects - angel (A) and blade (B). The results are in *mm* and the comparison is made between our proposed solution (Paper) and the built-in scaling solutions in ContextCapture (CC) and PhotoScan (PS)

	Paper (mm)	CC (mm)	PS (mm)
A	<b>0.35 ± 0.063</b>	0.48 ± 0.069	1.04 ± 0.093
B	<b>0.18 ± 0.012</b>	0.23 ± 0.008	0.56 ± 0.009

$$\sigma_{metric}^2 = D_{SfM}^2 \cdot \sigma_s^2 \quad (E.7)$$

Furthermore the scale uncertainty in *mm* can be also easily measured

through our proposed solution. We can take two random points from the unscaled measured object and calculate the Euclidean distance between them in unknown units -  $D_{SfM}$ . As we have calculated the scale uncertainty, denoted as  $\sigma_s$ , we can use Equation E.7 as given in the book by [24], to calculate scaled distance uncertainty  $\sigma_{metric}^2$  in the chosen metric. To demonstrate how this uncertainty can be useful, we recalculate the real world reconstruction's scale uncertainty, by using different number of position data - from 2 to 18. Each uncertainty is then used to find the metric uncertainty of the distance between the same two randomly chosen points. The results from the test can be seen in Figure E.8. The calculated metric uncertainty decreases with the introduction of more and more point positions.

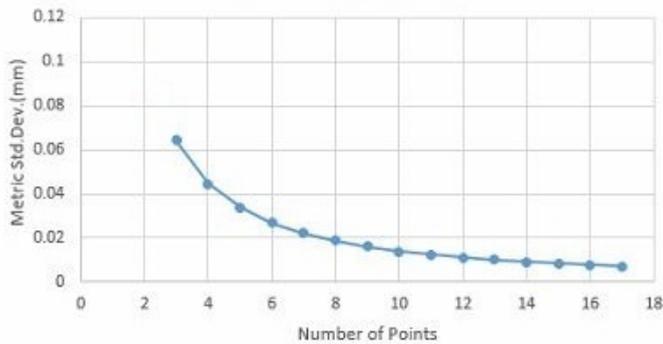


Fig. E.8: Correlation between the number of positions used to calculate the scale uncertainty and the metric uncertainty when measuring the real world distance between two points on a object

## E.4 Conclusion and Future Work

In our paper we presented a pipeline for computing the absolute scale of a 3D model reconstructed using SfM. Our method relies on using external positioning information from a GPS-RTK sensor, which has an inherent uncertainty present in the provided data. We provide an analysis of this uncertainty and how it propagates to the calculated absolute scale and results in a scale uncertainty. Through a series of tests we demonstrated how changes to the number of positions used and their spatial relationship can also influence the scale uncertainty. We tested two ways to find the scale uncertainty - an iterative method and a mathematical covariance propagation of noise method.

Finally, we tested our proposed pipeline against the scaling solutions available in state of the art SfM software solutions - ContextCapture and PhotoScan(Metashape). We demonstrate that we achieve better results, on

top of providing more information about the scaling uncertainty.

As an extension to the current paper, we propose testing the pipeline using data captured through drone flights. This way the GPS-RTK positioning information can be tested in different weather and environment conditions. Additionally the testing on objects with different sizes will provide data on how the method scales with size and if the uncertainty depends on the size of the scanned object. Finally, different positioning systems would also be tested and modeled - both indoor and outdoor, to make the pipeline more versatile.

## Acknowledgements

This work is funded by the LER project no. EUDP 2015-I under the Danish national EUDP programme. This funding is gratefully acknowledged.

## References

- [1] A. Corti, S. Giancola, G. Mainetti, and R. Sala, "A metrological characterization of the kinect v2 time-of-flight camera," *Robotics and Autonomous Systems*, vol. 75, pp. 584–594, 2016.
- [2] H. Sarbolandi, D. Lefloch, and A. Kolb, "Kinect range sensing: Structured-light versus time-of-flight kinect," *Computer vision and image understanding*, vol. 139, pp. 1–20, 2015.
- [3] M. Sarker, T. Ali, A. Abdelfatah, S. Yehia, and A. Elaksher, "a cost-effective method for crack detection and measurement on concrete surface," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, p. 237, 2017.
- [4] A. Jamaluddin, O. Mazhar, C. Jiang, R. Seulin, O. Morel, and D. Fofi, "An omni-rgb+ d camera rig calibration and fusion using unified camera model for 3d reconstruction," in *13th International Conference on Quality Control by Artificial Vision 2017*, vol. 10338, 2017.
- [5] J. Schöning and G. Heidemann, "Taxonomy of 3d sensors," *Argos*, vol. 3, no. P100, pp. 9–10, 2016.
- [6] C. W. VisualSFM, "A visual structure from motion system," 2011.
- [7] J. L. Schonberger, "Colmap," <https://colmap.github.io>, 2016, accessed: 2018-09-06.
- [8] Bentley, "Contextcapture," <https://www.bentley.com/en/products/brands/contextcapture>, 2016, accessed: 2020-06-15.

## References

- [9] Agisoft, "Metashape (photoscan)," <http://www.agisoft.com/>, 2010, accessed: 2020-06-15.
- [10] CapturingReality, "Reality capture," <https://www.capturingreality.com/>, 2016, accessed: 2020-06-15.
- [11] M. Rabah, M. Basiouny, E. Ghanem, and A. Elhadary, "Using rtk and vrs in direct geo-referencing of the uav imagery," *NRIAG Journal of Astronomy and Geophysics*, 2018.
- [12] M. Dugan, "Rtk enhanced precision geospatial localization mechanism for outdoor sfm photometry applications," *Robotics Research Journal*, 2018.
- [13] T. Schöps, T. Sattler, C. Häne, and M. Pollefeys, "3d modeling on the go: Interactive 3d reconstruction of large-scale scenes on mobile devices," in *3D Vision (3DV), 2015 International Conference on*. IEEE, 2015, pp. 291–299.
- [14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [15] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.
- [16] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *International workshop on vision algorithms*. Springer, 1999, pp. 298–372.
- [17] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 4, pp. 376–380, 1991.
- [18] M. Daakir, M. Pierrot-Deseilligny, P. Bossier, F. Pichard, C. Thom, and Y. Rabot, "Study of lever-arm effect using embedded photogrammetry and on-board gps receiver on uav for metrological mapping purpose and proposal of a free ground measurements calibration procedure." *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2016.
- [19] DJI, "D-rtk gnss," <https://www.dji.com/>, 2017, accessed: 2018-09-06.
- [20] I. Nikolov and C. Madsen, "Benchmarking close-range structure from motion 3d reconstruction software under varying capturing conditions," in *Euro-Mediterranean Conference*. Springer, 2016, pp. 15–26.

## References

- [21] R. M. Haralick, "Propagating covariance in computer vision," in *Performance Characterization in Computer Vision*. Springer, 2000, pp. 95–114.
- [22] C. B. Madsen, "A comparative study of the robustness of two pose estimation techniques," *Machine Vision and Applications*, vol. 9, no. 5-6, pp. 291–303, 1997.
- [23] D. Girardeau-Montaut, "Cloudcompare-open source project," *Open-Source Project*, 2011.
- [24] M. H. DeGroot and M. J. Schervish, *Probability and statistics*. Pearson Education, 2012.

# Paper F

## Calculating Absolute Scale and Scale Uncertainty for SfM using Distance Sensor Measurements

Ivan Nikolov and Claus Madsen

The paper has been published in the  
*Recent Advances in 3D Imaging, Modeling, and Reconstruction*, pp 168-192,  
2020.

© 2020 IGI Global  
*The layout has been revised.*

## Abstract

*Capturing details of objects and surfaces using Structure from Motion (SfM) 3D reconstruction has become an important part of data gathering in geomapping, medicine, cultural heritage and the energy and production industries. One inherent problem with SfM, due to its reliance on 2D images is the ambiguity of the reconstruction's scale. Absolute scale can be calculated, by using the data from additional sensors. This chapter demonstrates how distance sensors can be used to calculate the scale of a reconstructed object. In addition, the authors demonstrate that the uncertainty of the calculated scale can be computed and how it depends on the precision of the used sensors. The provided methods are straightforward and easy to integrate into the workflow of commercial SfM solutions.*

## F.1 Introduction

Structure from Motion (SfM) techniques have matured throughout the years to become viable commercial solutions for 3D reconstruction. This is due to the techniques' scalability, relative ease of use and the fact that they do not rely on specialized equipment. This positions SfM as a useful substitute for other reconstruction approaches that require both specialized hardware and software, like structured light [1], stereo [2] or time-of-flight cameras [3], when real-time performance is not necessary.

The algorithm pipeline for SfM is extensively documented by [4] and the accuracy of different solutions for varying use cases are discussed by [5, 6]. There are several approaches to performing SfM reconstruction, but a typical algorithm takes 2D images looking at the reconstructed object or surface, from different positions and directions. Another important feature of SfM is the possibility to use it both with images from precisely calibrated capturing setups [7], as well as with in the wild image datasets [8], requiring more post-processing in filtering the image data and clustering it, but saving on long capturing times.

In the SfM processing pipeline, a number of feature points are extracted from each image and matched with features from the input images. These feature matches are filtered and together with the intrinsic parameters of the cameras are used in a bundle adjustment algorithm to triangulate the camera positions in 3D space, as well as a sparse point cloud. A depth map and dense point cloud are then computed. Finally, if needed, the dense point cloud is meshed and a texture is calculated from the images. One drawback of using only uncalibrated 2D images as input is that the scale of the reconstruction is ambiguous. To calculate the absolute scale, additional information is needed. This information can be captured manually, by using objects of known sizes in the images or by using additional sensors.

This chapter focuses on using additional sensors for calculating the absolute scale of the 3D reconstruction. It demonstrates a step-by-step solution which uses external distance sensors to provide the necessary information. In addition, the authors take into consideration that real-world sensors' readings contain level of uncertainty, which in turn is transferred to the calculated scale. The discussed solutions take this into consideration and demonstrate that these uncertainties can be quantified.

The chapter's contributions to the field of SfM can be summarized as:

- A lightweight and easy to implement method for finding the absolute scale of a SfM reconstruction using distance sensors;
- The method is easy to integrate into existing commercial SfM solutions, as it requires only simple outputs, such as a 3D mesh and camera positions and orientations;
- The method is flexible and can be used both with expensive LiDAR solutions, as well as cheap distance measurement sensors, as it does not require capturing the object surface;
- The uncertainty of the computed absolute scale can be calculated, when high precision is required.

## F.2 State of the Art

All state of the art SfM solutions, both open-source [9–11] and commercial [12–14], contain some kind of built-in way to manually scale the final 3D reconstruction using information captured from the environment. Normally, users can manually measure parts of the real and reconstructed objects and compute the resultant scaling factor. Another widely used method is relying on markers with predefined shapes and sizes. These markers are put on the reconstructed object or surface and are captured in the images. Later the scale of the object can be calculated from the ratio between the real-world size of the marker and the captured size. Both methods rely on the fact that the real object is easily accessible and are time consuming, which makes them not ideal for all use cases. In addition, there is no easy way to predict any introduced uncertainty to the calculated absolute scale.

Other approaches for estimating the scale, base assumptions on known factors of the capturing environment, like the height [15, 16] or the kinematic model of the motion [17]. Finally, some approaches rely on data from external sensors. Positioning data is widely used for geo-referencing and scaling the reconstruction, as seen in the work by [18–21]. Other multimodal approaches [22, 23] take advantage of inertial measuring units (IMU) for calculating the positioning, rotation and movement of the cameras. Combining

SfM and LiDAR scans has the added benefit of determining the scale of the reconstruction, as shown by [24, 25].

Building on the findings of the state of the art, a lightweight scaling solution can be achieved by introducing additional sensor data. This way data for the scaling can be captured together with the image data without the need for additional manual measurements. Position sensors are widely used for outdoor SfM reconstruction, but their reliance on GPS and other external position sensors makes them impractical for indoor use. Another problem with these systems can arise if the captured object or surface moves, while images are taken, as the captured positions are normally of the camera. To mitigate this problem, the position of the captured object needs to be tracked as well, making the setup even more complicated. Movement tracking sensors like IMUs are another possibility, but they require filtering and constant calibration to offset the possible drifting. Finally, distance sensors like LiDAR solutions give high precision measurements, but most present systems try to create a full 3D scan of the environment, which can introduce considerable computation load.

The solution described in the chapter takes the idea of using distance measurements from the camera to the captured object or surface, to calculate the scale of the reconstruction, but minimizes the necessary readings. It does not need to create a full 3D scan and can be used with both single direction distance measurement sensors, as well as high-quality multi-directional LiDAR solutions. This can streamline the often long and complicated process of capturing additional data for SfM reconstructions and provide a simple scaling way to already existing 3D reconstruction pipelines [26].

Additionally, because all sensors possess a certain amount of noise in their readings, the solution takes that into account and computes the scale together with the amount of uncertainty. This way highly precise measurements can be taken from the reconstructions.

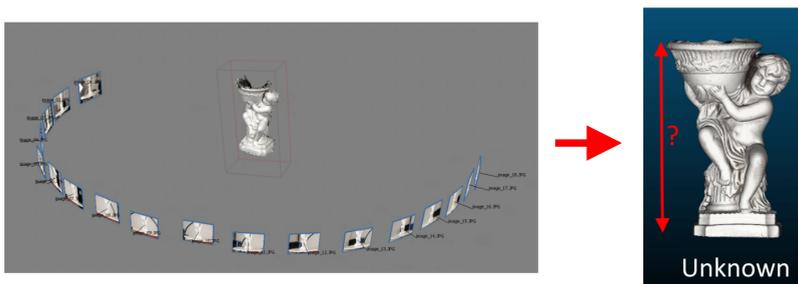
## **F.3 Methodology**

The methodology is divided into parts, containing the different steps that go into building the proposed solution. As an initial step, a basic SfM pipeline will be described and the production of a 3D reconstruction, with an unknown absolute scale, together with the other outputs needed for the proposed solution. Using this as a basis, the proposed solution of using distance sensors is introduced. For producing a 3D reconstruction with an absolute scale, two main parts are needed – a calibrated fixed rig containing the distance sensor and the camera used for SfM and an algorithm for comparing the real distance measurements, to the ones captured from the reconstructed objects or surfaces. Finally, when needing high degree of accuracy in the

calculated scaling factors, the uncertainty introduced by the used distance sensors needs to be taken into account. To do this, two additional parts are introduced in the proposed solution – an initial sensor uncertainty modeling and an algorithm for propagating this uncertainty to the calculated scaling factor.

### F.3.1 SfM Reconstruction Overview

For reconstructing objects or surfaces using SfM, only a normal color camera sensor is required. More importantly, images need to be taken of the scanned object, from varying positions and angles, so the overall shape, size and surface detail can easily be described. Images need to be taken with a certain degree of overlap, normally at least 70% to 80% overlap, as specified in the work by [4, 5]. From each image a number of feature points are extracted and matched between images. Feature extractors and descriptors like SURF [27], SIFT [28], FAST [29] or ORB [30] are used with a lot of commercial SfM, having proprietary solutions. From these matched features, together with the calculated intrinsic camera calibrations, a 3D triangulation is done between the unordered images, using any of the many bundle adjustment implementations [31]. This results in the creation of a sparse 3D point cloud, plus the calculation of the back-projected camera positions. The sparse point cloud can then be densified, by interpolating the information from the calculated 3D points. The resultant dense point cloud can be meshed if needed, with additional steps like texture calculation from projecting the camera images onto the 3D model, removing noise, closing holes, etc. If no additional information is provided to the application, the final output is scaled to an unknown scale, with the SfM applications trying to guess a correct one depending on the calculated camera positions, intrinsic parameters, size of the captured object, etc. This calculated scale can vary from one object to another and can be different even between multiple computations on the same datasets, as seen in Figure F.1.



**Fig. F.1:** Example of a SfM reconstruction, resulting in a 3D object with unknown absolute scale. Several images of the object are taken from different positions and angles

### F.3.2 Scaled SfM Reconstruction Using Distance Sensors

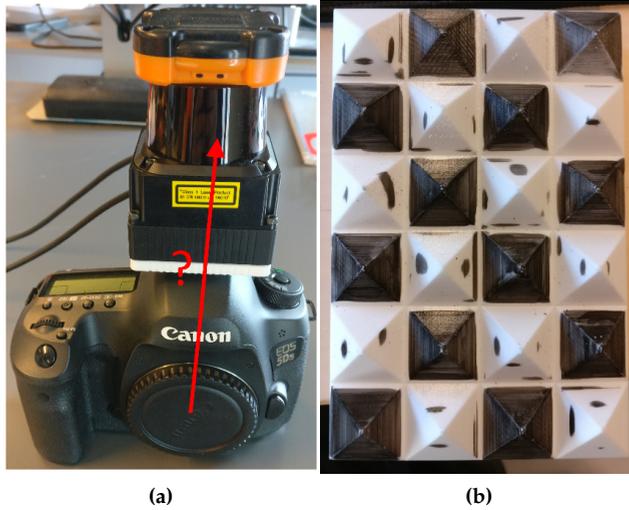
To introduce an absolute scale to the SfM pipeline additional real-world data is needed. In the method proposed by the authors this data comes from distance sensors. In the next chapter the basis for the proposed algorithm will be explained. For this chapter the sensor used as an example is the mid-range LiDAR solution by Hokuyo - the UTM-30LX [32]. As it will be shown in the later chapters and the Results section, this sensor can easily be interchanged with other cheaper alternatives and the proposed methods will retain their validity. Initially the chosen sensor needs to be attached to a fixed rig together with the camera, used for capturing images for SfM. The transformation between the two needs to be calibrated as well. This initial calibration needs to be done only once and after that the rig can be used together with the proposed algorithms for calculating the absolute scale of the captured objects.

#### Extrinsic Calibration Between Camera and Sensor

An initial extrinsic calibration between the real-world camera and each used distance sensor needs to be calculated. This calibration provides the position and orientation of the sensor compared to the camera, which eliminates problems that can arise from the lever-arm effect [33], which is normally present when using GPS sensors for georeferencing. It also gives an initial guess for the scale of the reconstructed object. It is important to note that in a perfect world this calibration would be enough to give a correct absolute scaling factor, but because of possible errors in the calibration and initial positions of the camera and sensor, here the authors use it only as an initial guess, which is used as a basis for the second part of the proposed method. Also, this part needs to be done just once as long as the transformation between the sensor and the camera is not changed. The proposed calibration steps are the same for any used distance sensor.

To prevent any unintended movements, the sensor is mounted on a bracket together with the camera with the forward direction of the two pointing at the same direction, as seen in Figure F.2a. The position and orientation between the two is unknown and the proposed calibration is done using a 3D checkerboard, consisting of a flat plane with a set of 24 four sided pyramids, as shown in Figure F.2b. This design is chosen over a completely flat one, as it provides depth resolution for the readings on the distance sensor.

Here the required coordinate systems need to be explained. First there are the distance sensor readings' 2D local coordinate system, which are represented as polar, using angles and distances. Next are the local coordinate systems of the reconstructed cameras, each with their orientation. Finally, there is the world coordinate system of the 3D reconstructed object, in which



**Fig. F.2:** The 3D checkerboard calibration artifact, used for finding the transformation between the sensors and the camera.

both the cameras and the object are position. To achieve a proper calibration, the sensor readings' 2D local coordinates, need to be expressed as in the world coordinate system of the reconstruction. To perform the calibration between the cameras and the sensors, each needs to be able to produce a 3D view of the checkerboard.

For the capturing camera's view, the 3D model is produced from a SfM reconstruction from multiple images. To ensure a correct reconstruction both horizontal images in a semi-circle and vertical images are taken, with distance sensor readings taken together with the vertical images. A SfM 3D point cloud is produced from these images.

Next the same needs to be done from the distance sensor's readings. Because the sensor's readings are in its local coordinate system, they need to be transformed into the unified coordinate system of the SfM reconstructed point cloud. SfM solutions calculate the camera positions together with the reconstructed objects. For simplicity, the authors will show the calculation for one camera, from the full set  $k = [1, 2 \dots C]$ . same calculations are valid for all other cameras.

The camera position is used as an origin point to transform the distance sensor's readings to the proper coordinate system. To do this first, the sensor readings are transformed from polar to Cartesian coordinates, with the camera position as origin, using Equation F.1, where in the case of the HC-SR04, the direction angle of the only distance measurement is set at 0 degrees. In the equation,  $x_{origin}, y_{origin}, z_{origin}$  are the coordinates of the camera

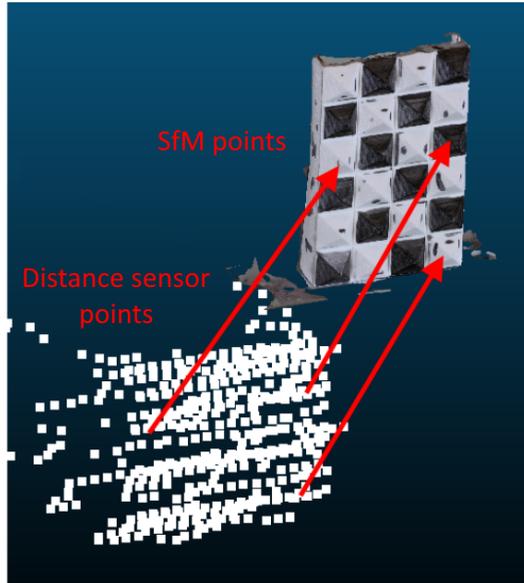
### F.3. Methodology

position origin,  $l_i$  is the sensor distance at angle  $\phi_i$  for each of the readings, where the number of readings can vary and for simplicity it is expressed as  $i = [1, 2, \dots, n_k]$ .

$$\begin{aligned} x_i &= x_{origin} - l_i \cos \phi_i \\ y_i &= y_{origin} - l_i \sin \phi_i \\ z_i &= z_{origin} \end{aligned} \quad (\text{F.1})$$

The transformed sensor point set is then rotated in 3D space from the camera's local coordinate system to the global coordinate system of the reconstruction, using the camera's calculated rotation matrix, denoted in Equation F.2 as  $R$ . The SfM dense point cloud and transformed Cartesian readings from the UTM-30LX LiDAR can be seen in Figure F.3.

$$\begin{bmatrix} x_{i_w} \\ y_{i_w} \\ z_{i_w} \end{bmatrix} = R \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \quad (\text{F.2})$$



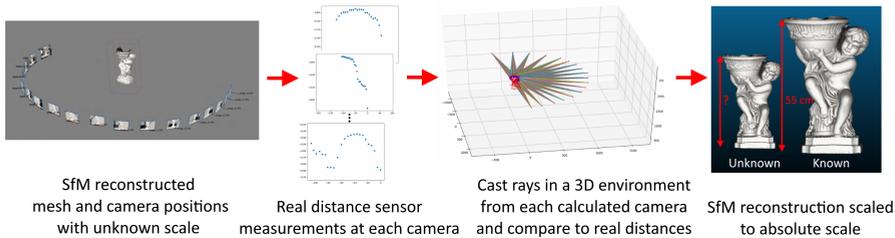
**Fig. F.3:** The two-point clouds used for the initial calibration process for detecting the transformation between the camera and sensor. Point pairs are manually selected from the dense SfM point cloud and the sparse sensor point cloud and aligned. Here the points captured from the UTM-30LX LiDAR are shown, with other sensors, using the same method for calibration

Once the 3D point cloud, created from combining all the distance readings, is in the same coordinate system, as the one created from the SfM re-

construction, the final transformation between them can be done. This transformation finds the position of the distance sensor's origin compared to the capturing camera, at each position. There are a number of manual and automatic ways to achieve this, but because there is no direct correspondence between the points of the two point clouds, the authors use the manual cloud alignment tool, which comes with CloudCompare [34]. To calculate the transformation matrix, a number of equivalent points are selected from the two point clouds and the distance error between them is minimized. As a result, a transformation matrix is calculated, which is used in the next part of the method.

### Absolute Scale Calculation Using Distance Measurements

Once the initial calibration is done once, the next steps can be done for each measured object or surface. For each camera position, a sensor reading of the distance to the object is taken. These distance readings are used as input for the proposed solution, together with the reconstructed SfM object and the camera positions and orientations. The process is shown in Figure F.4 and explained in the next paragraphs.

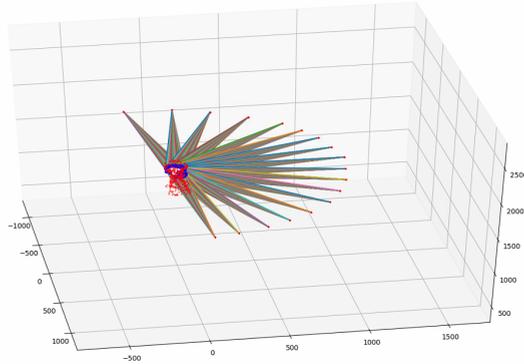


**Fig. F.4:** Second part of the solution for finding the absolute scale of a SfM reconstruction using the ratio between real-world and calculated distance measurements.

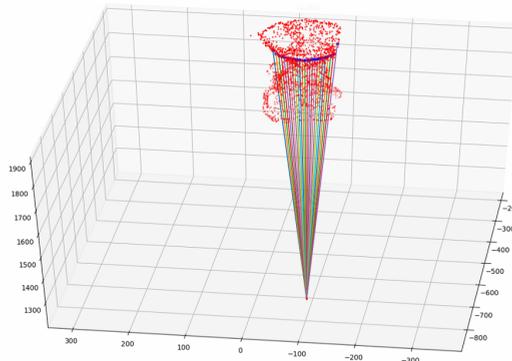
To calculate the absolute scale of the SfM reconstruction, the real-life sensor measurements from each camera position need to be compared to the distances calculated from reconstructed camera position to the reconstructed object or surface. Naturally, in a best-case scenario these two distances can be the same, but because of the way SfM calculates the reconstructed object, as well as imperfections in the calibration, etc., the two distances can be far off. Initially each camera position is taken and the inverse transformation matrix, calculated in the initial calibration part, is used to find the position of the distance sensor for that camera. A ray is cast from the distance sensor's position toward the reconstructed object and if they hit it, the distance between the two is calculated and saved. The number of rays cast from the position of the sensor for each camera depends on the angular resolution of that sensor. To demonstrate the idea, in Figure F.5a the rays that hit the reconstruction from

### F.3. Methodology

one sensor position are shown, while in Figure F.5b, the same is shown for a number of sensor position. The two figures show calculated distance rays created using the characteristics of the UTM-30LX LiDAR.



(a)



(b)

**Fig. F.5:** Visualization of the rays from a single camera and the points hit on the object (Figure F.5a), together with the ray hits from each camera on the reconstructed mesh (Figure F.5b). UTM-30LX LiDAR readings. For easier visualization only a subset of the vertices of the mesh is visualized

To simplify the calculations, an average distance  $\bar{r}$ , is computed from all the calculated distance rays  $r_j$ , for each camera position. The number of distance rays  $j = [1, 2, \dots, m_k]$  can vary from camera to camera, depending on the shape of the reconstructed mesh and is denoted with  $m_k$ .

To calculate the scale factor between the real-life object and the reconstruction, real sensor readings are needed from each camera position. Here it is important to note that a hypothesis is made that for each camera position that the real sensors have readings on the object, the same camera position

also has calculated ray distances from the reconstructed object mesh. With this the number of cameras is denoted by  $k = [1, 2 \dots C]$ , where  $C$  is the number of camera positions with readings both from the real and reconstructed object. The real-life sensor measurements are denoted in the same way, as the one used for the extrinsic calibration -  $l_i$ , where  $i = [1, 2, \dots n_k]$ . Here again the readings can vary from camera to camera and for simplicity they are marked with one variable. From all the sensor readings of each camera an average distance measurement is calculated  $\bar{l}$ . From the two average distance readings – the real life one and the reconstructed one the scaling factor  $s_k$  can be calculated as a ratio for each camera position. An average scaling factor is calculated from the factors calculated for each camera position. This calculation is presented in Equation F.3. Here is important to note that if a single direction distance sensor is used, like the HC-SR04, then  $n_k = m_k = 1$ , for each camera and the average values of both the real and the calculated distances, can be substituted for the single reading.

$$s = \frac{1}{c} \sum_{k=1}^c \left( \frac{\frac{1}{n_k} \sum_{i=1}^{n_k} l_i}{\frac{1}{m_k} \sum_{j=1}^{m_k} r_j} \right) = \frac{1}{c} \sum_{k=1}^c \left( \frac{\bar{l}_k}{\bar{r}_k} \right) \quad (\text{F.3})$$

Normally one pass should be sufficient to calculate a proper scale factor, but there are a number of complicating factors, which can make this calculation incorrect:

- Imperfections in the initial extrinsic calibration – because the initial calibration requires manual input and uses a distance minimization algorithm, there is always a possibility of falling into a local minimum, not representative of the best possible solution of the transformation;
- Captured objects or surfaces can have complex shapes with both micro- and macro-roughness, which can present problems for both the real distance sensor's readings, as well as for the distance readings from the reconstructed object. Small changes in the direction of the cast ray can give large changes in the returned distances. This can be mitigated with the use of distance sensors with higher angular resolutions in both 2D and 3D, but it is still a large factor for less complex sensors;
- Because of the way SfM works, by matching features from multiple 2D images, it is prone to creating surface noise and reconstruction inaccuracies, when the capturing conditions are sub-optimal. These problems can be mitigated somewhat with the use of surface smoothing and denoising algorithms, but can still interfere with the calculated distances from the reconstructions, throwing off the initial computed scaling factors.

### F.3. Methodology

A way to deal with these possible problems is to do multiple iterative passes on calculating the scaling factor. Before each pass the previously calculated factor is used to scale the reconstruction. Rays are then cast again and a new factor is calculated. The algorithm stops either after a predetermined number of times or when the calculated Root Mean Square Error (RMSE) between the real sensor distances and the calculated ones from the reconstruction drops below a certain threshold. Once the end criteria are reached, the final scaling factor  $s_{final}$  is calculated, by multiplying all the temporary factors  $s$ , as seen in Equation F.4, where  $g$  is the number of iterations needed to reach the end criteria. Here it is important to note that the average sensor distance  $l_{avg}$  is measured in  $[mm]$ . In the first iteration of the algorithm the ray distances are unitless  $[\cdot]$ , while in the consecutive iterations they are again in  $[mm]$ . This means that the final unit of the scaling factor is also  $[mm]$ .

$$s_{final} = \prod_1^g s \quad (F.4)$$

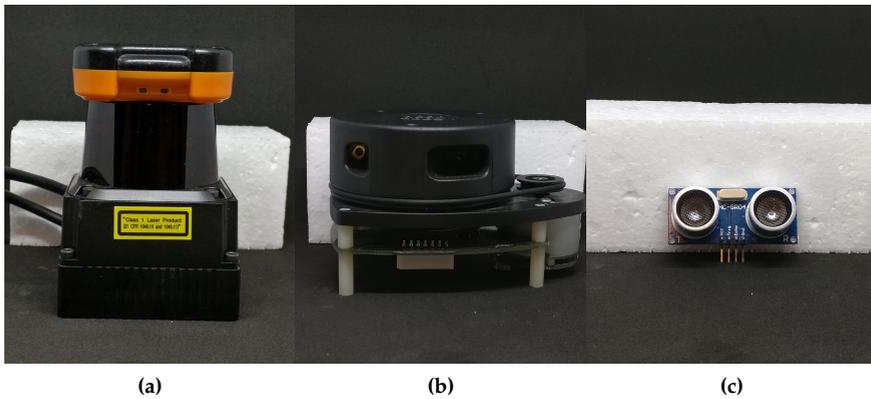
For most SfM reconstruction this can be a final result, which would be useful for detecting the absolute scale in a simple and straightforward way, with minimal additional captured data. But when a high degree of precision is required from the reconstruction, for example for surface inspection, where millimeter or even sub-millimeter accuracy is the norm, additional steps are required. Each used sensor has inherent noise and uncertainty in its readings and this uncertainty can propagate to the calculated scale, making data measurements from the reconstruction more unreliable. This problem has been expressed in the paper by [21], using the notion that uncertainties from one type of variables, in this case scale, can propagate to another type of measurements, for example surface measurements. In Equation F.5 taken from the paper, the uncertainty of a calculated scale factor is denoted as its variance  $\sigma_s^2$ . The measurement  $D_{SfM}$  in unknown scale is taken from the surface of a 3D object and subsequently scaled to absolute scale using that scale factor. Then it will display a measurement uncertainty  $\sigma_{metric}^2$ , proportional to the scale factor's uncertainty. If the scaling uncertainty is known, then this can be predicted and better-quality measurements can be achieved.

$$\sigma_{metric}^2 = D_{SfM} \cdot \sigma_s^2 \quad (F.5)$$

The next chapters look into modeling the distance measurement sensors' uncertainties and propagating this uncertainty through the algorithms used for calculating the absolute scale factors.

## F.4 Uncertainty Modeling and Propagation

To demonstrate the uncertainty modeling of a distance sensor, a number of hardware solutions are chosen. These solutions range in price, the type of distance data they capture and their performance. The three sensors are a low-cost ultrasonic distance sensor HC-SR04, a low-cost LiDAR solution - the rpLidar-A1 [35] and the already mentioned Hokuyo UTM-30LX. Figure F.6 shows the sensors, as well as examples of the captured uncertainties from each one. A more in-depth explanation is given below. Initially their distance capture uncertainty is calculated, which will later be used as an input in the propagation algorithm.



**Fig. F.6:** The distance sensors used for testing the proposed solution – ultrasonic distance sensor HC-SR04 (Figure F.6a), LiDAR rpLiDAR-A1 (Figure F.6b) and LiDAR Hokuyo UTM-30LX (Figure F.6c)

### Distance Sensor Modeling

The three used sensors provide different measurement resolutions and uncertainties, as well as maximum measured distances. The HC-SR04 provides a single measurement and can measure between 0.1-4m, the rpLidar-A1, can measure between 0.5-12m and has an angular resolution of around 1 degree, the UTM-30LX can measure between 0.5-30m and has an angular resolution of 0.25 degrees. The distance measurement uncertainty of the sensors at multiple distances needs to be captured for use in the scale uncertainty calculation. To do this, readings of each sensor are taken in intervals of 0.2m, from 0.5m to 3.1m. The bottom interval is chosen, because of the limitations of the tested sensors. The top interval is chosen, because of the size of the laboratory where the tests were conducted. A ground truth is taken for all the

#### F.4. Uncertainty Modeling and Propagation

readings using a Leica DISTO laser distance meter, with a known accuracy of less than 0.03m. A matte, flat and smooth surface is used for taking the readings to minimize errors from laser beam reflections. For the two LiDAR units, only the central beam is taken into consideration, for simplification of the measurements. At each distance 500 readings are taken. Both the distance error between the ground truth and the measurement and the standard deviation between the 500 measurements are calculated. These are shown in Figure F.7 and Figure F.8. To verify that the method for capturing the readings is correct, the standard deviation of the measurements captured by [36] for the similar Hokuyo UST-20LX are plotted as a comparison. It can be seen that the authors' results exhibit similar standard deviation to those. To create the uncertainty values used in the second part of the proposed solution, the standard deviation measurements for each of the devices are interpolated and smoothed using a spline fitting. The created values are used as a look-up table depending on the capturing distance. The distance errors are also used to correct the captured measurements when using the sensors.

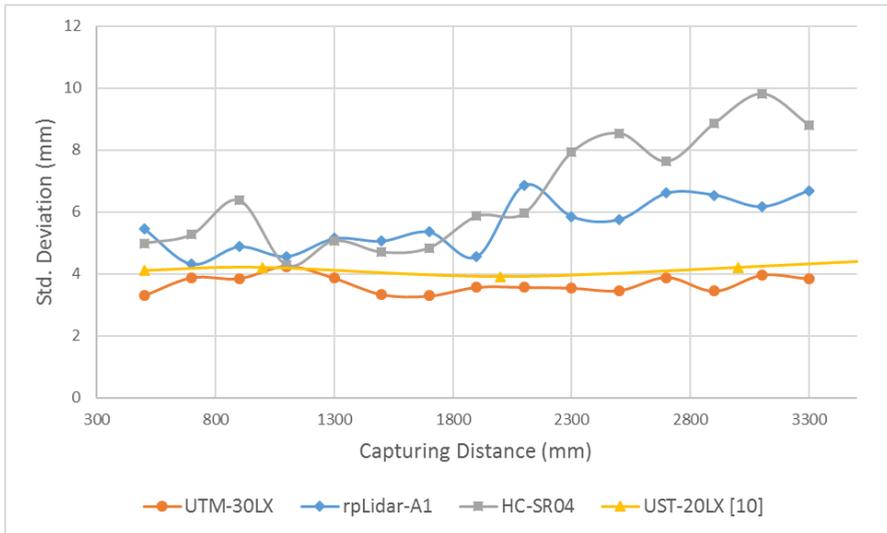
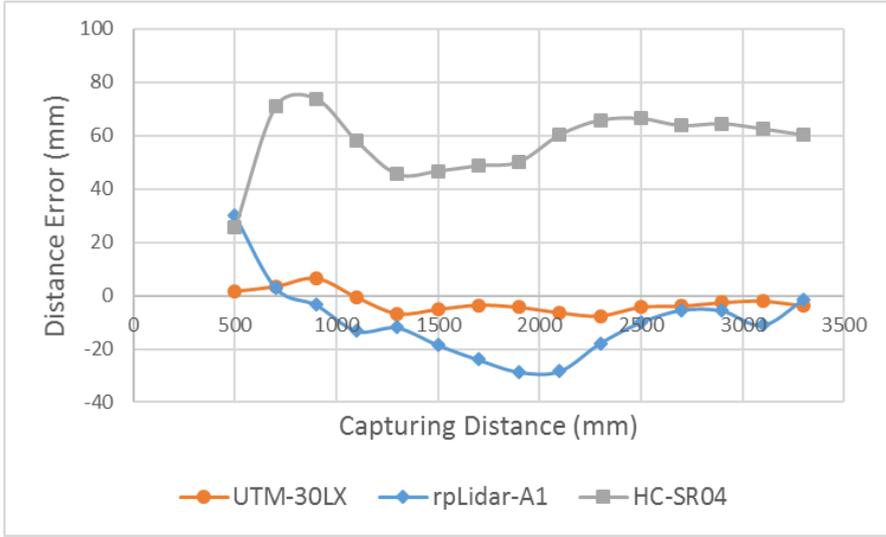


Fig. F.7: Captured distance standard deviations from the tested sensors - UTM-30LX, rpLidar-A1 and HC-SR04, together with the results taken from [36] for comparison

#### Propagation of Distance Uncertainty to Scale

As the final scaling factor is calculated, the next important thing to take into account is the possible noise and uncertainty, that the used sensor might have introduced to it. The authors' proposed solution is based on the described approach for covariance propagation of noise for computer vision as described



**Fig. F.8:** Captured distance errors from the tested sensors - UTM-30LX, rpLidar-A1 and HC-SR04. The distance errors are calculated from as a difference between the captured measurement and ground truth measurements.

by [37]. This method is chosen as it has been demonstrated to give good results in the work by [38] and [21], as long as the input parameters used are independent from one another and are used in a function to calculate the output parameters, whose uncertainty needs to be found. The method works for both analytically and iteratively computed functions. In the work by [21] the method is used for calculating the uncertainty of the calculated scale, when using position measurements from a GPS. In that case the position in each dimension is used as a separate input parameter. To use the same numerical approach with distance measurements, the proposed equations needs to be simplified, so it can be used with only one input parameter. Equation F.6 described in the paper, can be used with the data produced by the authors' proposed solution, but the differentiation needs to be calculated only for the average measured distance from each camera position.

$$\sigma_s^2 = \frac{\partial s}{\partial l} \Delta \frac{\partial s^T}{\partial l} \quad (\text{F.6})$$

To do this the modelled sensor variance is subtracted and added to the average sensor measurements for each camera in turn and the scale is calculated for each, as seen in Equation F.7. The other part of Equation F.6 that needs to be changed is the calculation of the covariance matrix, again as there is only one input parameter. The new matrix is shown in Equation F.8, where the average sensor uncertainty is the calculated one from Equation F.9.

#### F.4. Uncertainty Modeling and Propagation

$$\frac{\partial s}{\partial l} = \left[ \frac{\partial s}{\partial l_1} \cdots \frac{\partial s}{\partial l_C} \right] \quad (\text{F.7})$$

$$\Delta = \begin{bmatrix} \sigma_{l_1}^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{l_C}^2 \end{bmatrix}_{C \times C} \quad (\text{F.8})$$

$$\sigma_{l_k}^2 = \frac{1}{n_k} \sigma_l^2 \quad (\text{F.9})$$

If the captured sensor readings for each camera positions are used as random variables, then they will contain the already extracted uncertainty, in the form of variance, which is denoted as  $\sigma_l^2$  for each reading. In Equation F.9, the average variance ( $\sigma_{(\bar{l}_k)}^2$ ) for the average sensor distance for each camera  $k = [1, 2 \dots C]$ , is taken from Equation F.3, where only the real sensor's data contains uncertainty. From the average of the real distance sensor's readings the variance can be represented using Equation F.10, based on the theorem 2 from [39], describing representing the variance of a linear function and a variable, in the current case that variable is the number of measurements per camera,  $n_k$ . From there the average variance can be represented using theorem 4 [39] as a sum of all the variances of all measurements per camera (Equation F.11).

$$\sigma_{l_k}^2 = \frac{1}{n_k^2} \sigma_{\sum_{i=1}^{n_k} l_i}^2 \quad (\text{F.10})$$

$$\sigma_{\sum_{i=1}^{n_k} l_i}^2 = n_k \sigma_l^2 \quad (\text{F.11})$$

By using Equation F.6 and taking two simplifications into account, the scale uncertainty can be computed in a more straightforward manner:

- The measured distances from the cameras to the object should be roughly the same for all images;
- The number of distance readings per camera from the real world and the cast rays to the reconstructed object needs to be the same.

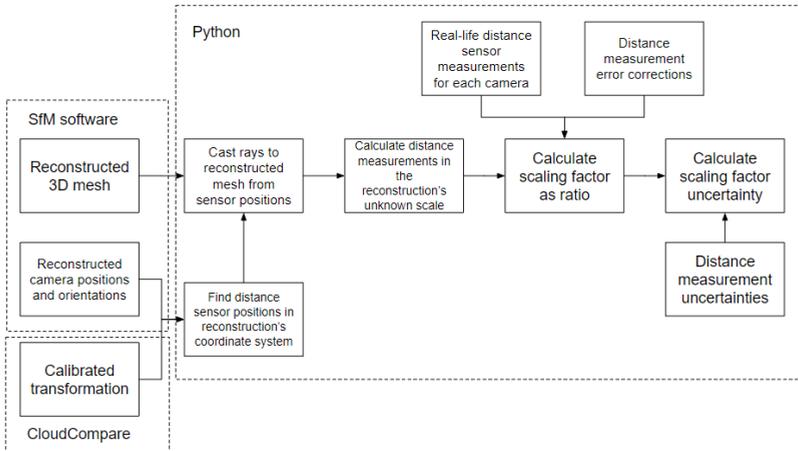
The first simplification can be restrictive, but in the work by [5] and [40], it is demonstrated that for achieving the best results in SfM reconstruction, a circular or hemispherical capturing path is needed. For this type of capturing the distances can be relatively uniform. The second simplification can also be followed if a set number of measurements are selected depending on the distance sensor and the same number is set for the calculated measurements from the reconstructed object. Taking this into account and expressing that

$n_k = m_k$  and  $f = \bar{r}_k$ , is the average distance measurement for all cameras  $k = [1, 2 \dots C]$ , then a new simplified expression can be made, as shown in Equation F.12.

$$\sigma_s^2 = \frac{1}{n^2 c f^2} \sigma_l^2 \quad (\text{F.12})$$

This expression gives a number of relations between the used variables. From the above, it can be seen that the scale uncertainty drops with both the number of camera positions with sensor measurements and the number of measurements per positions. This can be explained by the fact that the more positions there are, the more information for the distances between the cameras and the object's surface there is. Also, the more measurements there are per camera position, the less likely it is that the sensor's uncertainty will affect. In addition, the distance from the camera to the object can affect the uncertainty in a number of ways. If measuring large distances, a small millimetre sensor uncertainty will not give a lot of weight to the scale calculation, but if the reconstruction is done using images and measurements close to the object the distance uncertainty will be more pronounced. The proposed uncertainty solution is tested in the Results section and compared to two additional uncertainty calculation solutions.

## F.5 Implementation



**Fig. F.9:** Pipeline of the proposed solution. The bulk of the calculations are done using Python, with the needed inputs coming either from the distance sensor or as output from the chosen SfM software.

An overview of the implementation of the proposed solution is given in Figure F.9. The initial data comes as an output from the chosen SfM solutions. Because the reconstructed object or surface is going to be used for casting rays and measuring distances, a mesh is suggested. The calculated camera positions and orientations are also used as input for the proposed solution. These inputs are used in the main implementation together with the calibrated transformation between the camera and the sensor in the rig. The used algorithms are built in Python, together with the library trimesh [41], used for the implementation of casting rays and calculating the distance between their origin and the hit point. The position of the distance sensor is calculated for each camera location using the calibration transformation matrix and the position and orientation of the camera. Once the location and orientation of the sensor is known, a number of rays are cast from it. The number and density of the rays depends on the chosen sensor and set input parameters – single or multi distance measurement sensor, angle resolution and maximum working distance. The distances of all rays hitting the reconstructed object are captured. The real-life sensor measurement for each camera, corrected using the distance error expectation is used together with the ray distance to calculate a scaling factor. The average scale factor for all camera positions is calculated. This scaling factor is used to scale the reconstruction and camera positions and the process is repeated until the difference between the real measurement and the ray one is minimized. The sensor measurement uncertainty is calculated using Equation F.12, with the average number of measurements and the average distance between the cameras and the reconstructed object computed as part of the scale calculation.

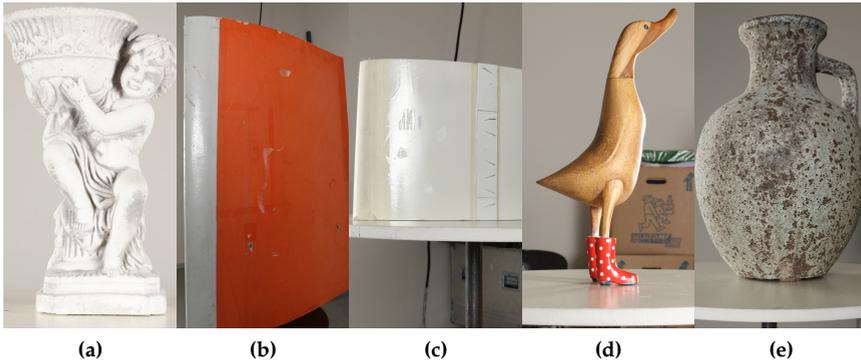
## F.6 Testing and Results

To test the performance of the proposed solution, a number of testing scenarios are created, using each of the three distance sensors. Five distinct objects with different shape, sizes and surface characteristics are chosen. To provide a comparison of the proposed method's performance, the authors compare it to two existing scaling solutions that rely on camera position data.

### F.6.1 Testing Setup

The five objects are selected to represent different 3D reconstruction scenarios - for both the industry and cultural preservation. The chosen objects can be seen in Figure F.10. Two of the objects are wind turbine blade pieces. Scanning wind turbine blades for detecting damages, by unmanned aerial vehicles (UAVs), has become widely used in the energy industry and a lot of research has been done on it [42, 43]. In addition, this type of scenario would

require a sensor-based way to scale the computed SfM reconstructions, as it is normally hard to manually measure the real-life blade surfaces or to place markers on them. The other three objects represent examples of digital cultural heritage preservation. Scanning fragile or hard to reach objects is another common use of SfM reconstruction, where access can be restricted and neither manual or marker-based scaling can easily be used. In addition, the five objects were chosen based on widely varying surface profiles and shapes, which can be used to judge the robustness of the proposed method.



**Fig. F.10:** The five objects were selected to represent two types of application scenarios, which may require the use of SfM reconstruction with absolute scale. The first group represents artifacts for digital cultural preservation - an angel statue (Figure F.10a), a duck statue (Figure F.10b) and a vase (Figure F.10c). The second group represents surface inspection for the industry, with examples of two types of wind turbine blade, denoted as the small one (Figure F.10d) and large one (Figure F.10e)

Initially, a number of images are taken from each of the five objects, in a 180-degree semi-circular pattern. This method of capturing the image is used, as it has been shown by [5, 44] that it produces good reconstruction results with the minimum needed image positions. For each image a sensor measurement is also taken. A Canon 5Ds DSLR camera is used with a 30-105 mm zoom lens and the taken images are with a resolution of 8688x5792. This way the possibility of inaccuracies on the reconstruction can be minimized and the only possible error can come from the scaling methods. Once the images are taken, a reconstruction is made of each object using Agisoft Metashape.

## F.6.2 Scale Test Results

The method presented in this chapter is compared to the built-in solution present in Metashape and the positioning-based solution used by [21], from here on referred to as NM. The built-in solution from Metashape, takes the

## F.6. Testing and Results

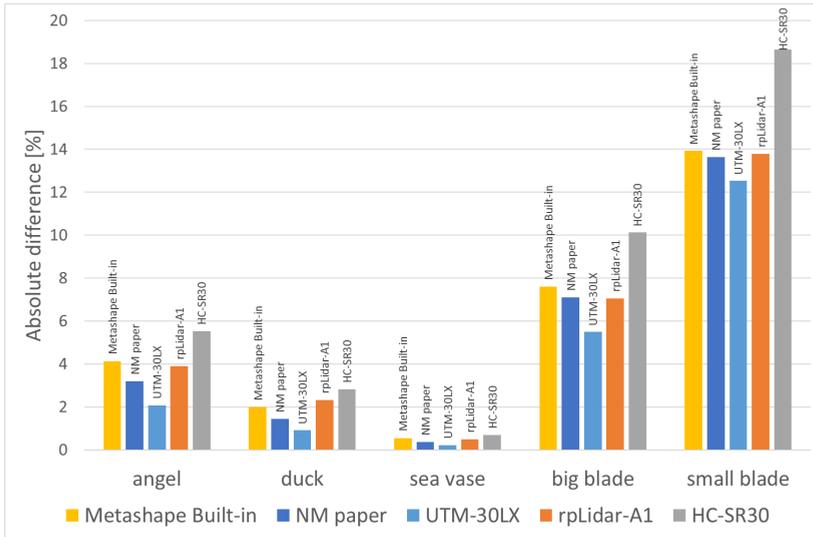
positioning data as input and as it reconstructs the object, it tries to use the data as initial guess where the cameras are. This has the effect that it also scales the model, but the used algorithm itself is proprietary and undisclosed by the company. No possible scaling uncertainties are calculated as well. The NM method uses positioning data for each camera. The data is captured from a GPS, with a real-time kinematic (RTK) add-on, which makes the positioning information much more accurate, compared to the normal GPS. A least-squares estimation is then used to calculate the transformation between these positions and the ones calculated by the SfM algorithm. The method makes the assumption that for each measured position there is a corresponding calculated one, without outliers. For testing purposes, the GPS-RTK sensor uncertainties in each direction are taken as they are and used in the present chapter. Because both the built-in solution and the NM one requires camera positions and the testing scenarios for these chapters are conducted indoor, the positioning data is captured manually, using the same laser range finder used for capturing the distance sensor uncertainty. In each direction the readings of the range finder are calculated and put in a global coordinate system, contained in the room where the experiments are conducted. As an added benefit this means that the captured positioning data is much more accurate than possible with a real positioning sensor like a GPS. In turn this will ensure that the two methods produce the best possible results for comparison to the proposed solution.

**Table F.1:** Absolute percentage difference between the averaged real-world and scaled 3D object measurements. For the proposed method three different sensors are used and for comparison the position-based built-in solution in Metashape and the solution by NM are tested. The two LiDAR sensors provide better or similar scaling performance than the position-based solution, while the low-cost single distance sensor still produces usable scaling factors.

Object	Metashape [%]	NM [%]	UTM-30LX [%]	rpLidar-A1 [%]	HC-SR04 [%]
Angel	4.12	3.20	2.07	3.89	5.53
Duck	1.99	1.44	0.90	2.33	2.82
Vase	0.55	0.36	0.22	0.50	0.69
Small Blade	7.60	7.09	5.49	7.06	10.12
Big Blade	13.92	13.64	12.52	13.79	18.63

The reconstruction is first created in Metashape, without an absolute scale. It is then scaled with all three tested solutions. To compare the accuracy of each of the calculated scale factors, a number of ground truth measurements need to be done. A number of parts of each of the real objects are manually measured with a digital calliper with an accuracy of  $0.02mm$ . The same parts are measured on the reconstructed objects after they have been scaled with the scaling factor from each of the methods. To minimize the possible

effect of human error from the manual measurements, each part is measured multiple times and an average measurement is calculated. To measure how close, the scaling from each method is to the real-life scale of the objects the absolute percentage difference between the real and the scaled measurements is calculated. The absolute percentage differences for each part of the object are then averaged for the whole object and a final scaling accuracy metric is calculated. The results from that can be seen in Table F.1, with graphical representation in Figure F.11.



**Fig. F.11:** The absolute percentage difference between the average real-world and 3D object measurements for each of the reconstructed objects. The proposed solution is tested with each of the distance sensors, together with the Metashape build-in solution and the method proposed by NM.

### F.6.3 Scale Uncertainty Test Results

The calculated scale uncertainty is also tested for each of the tested distance sensors from the proposed solution. For inputs each of the distance sensors, uses the calculated distance uncertainty lookup table. The results from the final scale factor uncertainty are given in Table F.2.

To test if the proposed simplified scale uncertainty solution gives proper results, it needs to be compared to other methods for calculating uncertainty. The work from [21], describes two different ways for introducing and computing scale uncertainty – a repeated approach and the numerical differentiation approach, already explained in the Methodology Chapter F.3. Each of the two methods is implemented with distance measurements as the in-

## F.6. Testing and Results

**Table F.2:** Calculated scale uncertainties for each of the five tested objects, with each of the distance sensors.

Object	UTM-30LX [ $mm^2$ ]	rpLidar-A1 [ $mm^2$ ]	HC-SR04 [ $mm^2$ ]
Angel	2.86e-10	1.01e-08	3.52e-08
Duck	5.79e-10	5.02e-08	2.90e-08
Vase	1.84e-10	4.34e-09	3.03e-08
Small Blade	1.04e-10	6.21e-09	1.25e-08
Big Blade	1.42e-10	2.65e-09	5.51e-08

put parameter and their results are compare to the ones produced by the simplified calculation. The angel statue has been chosen as a test object.

### Repeated Approach

The most straightforward way to compute the scale uncertainty from the distance sensor uncertainty is through repeated calculations. A zero mean Gaussian distribution is created from the standard deviations, obtained from the sensor uncertainty modelling. This distribution is then randomly sampled and values from it are added to the real sensor measurements for the 3D reconstruction. The scale is then calculated as normal and saved. This process is done repeatedly 1000 times, by randomly sampling the measurement fluctuations. Once all the final scale factors are calculated, their variance is calculated. This approach is easy to implement, but requires a lot of computation time to achieve good results, as well as a large pool of sensor measurements.

**Table F.3:** Comparison between three methods for calculating the scaling uncertainty – simplified model demonstrated by the authors and two methods described by [21] – a numerical model using covariance propagation and a repeated calculation model.

Sensor	Simplified [ $mm^2$ ]	Numerical [ $mm^2$ ]	Repeated [ $mm^2$ ]
UTM-30LX	2.86e-10	3.14e-10	2.95e-10
rpLidar-A1	1.01e-08	1.15e-08	1.05e-08
HC-SR04	3.52e-08	3.53e-08	3.65e-08

### Results

The scale uncertainty for each of the three sensors is calculated for the angel statue. The results from each of the three used methods are given in Table F.3. As it can be seen from the table, the simplified model demonstrates

discrepancy in the produced results for the sensors, which have higher uncertainty, combined with a smaller number of measured distances. The other two uncertainty computation approaches give comparable results.

#### **F.6.4 Discussion**

Analysing the results from the testing, it can be seen that the authors' proposed solution using the UTM-30LX LiDAR gives the best scaling results, together with the lowest scale uncertainty. This is an expected result, as the sensor has the highest angular resolution, together with the lowest distance measurement uncertainty. The other distance sensors also give progressively worse results depending on their distance measurement accuracy and the angular resolution. More interesting are the results from the HC-SR30 distance sensor. Even though the sensor has a comparatively small measurement distance and low distance accuracy, it manages to provide satisfactory results in all cases, except the second wind turbine blade. All tested algorithms show difficulties in scaling the wind turbine blades. It is theorized that this can be a problem from the reconstruction itself, as both objects are relatively harder to reconstruct and have noise in the reconstructed 3D models, as well as in the positioning of the cameras. This can result from glossiness of the surfaces and the lack of features. Additionally, the two blades have elongated shapes, which could additionally present problems, especially for the single distance measuring sensors, while with the help of more measuring points the two LiDAR solutions give a better average representation.

Looking at the scale uncertainty calculation, the same trend can be seen. The images with distance reading from the three sensors are taken from roughly the same positions and the same number of camera positions are used. From Equation F.12, that leaves only the number of distance measurements per camera position and measurement uncertainty as changing variables. It can be seen that, because the sonic sensor relies only on one reading and has the highest uncertainty, the produced scale from it also exhibits the highest uncertainty. On the other hand, the UTM-30LX sensor produces a lot of distance readings and together with the relatively smaller uncertainty, it achieves lower scale uncertainty values. That said, the cheap sensor still manages to produce good results and is viable for use.

When comparing the proposed simplified method to the other two computationally more expensive methods, comparable results are achieved. This shows that simplified method can be a useful approximation, when the longer computational times of the other method are prohibitive. More tests need to be conducted to get a predictive trend how the simplified method's results will behave with different distances, number of measurements and sensors.

Overall the proposed method gives better or comparable results to the two solutions using positioning sensors. Combined with the robustness of

a distance sensor, which can be used both indoor and outdoor and in varying conditions, the fact that the method does not require additional external hardware, and the flexibility that switching to a cheaper, but lower resolution sensor still gives usable results, shows the validity of the proposed solution.

## F.7 Conclusion

One of the problems connected to SfM is the inability to compute the absolute scale of a reconstructed object, without additional information. This chapter proposed a method for finding the absolute scale and scale uncertainty, by using the readings of distance sensors. The method can be made to work with a wide array of sensors and can easily be integrated into a SfM working pipeline, as it requires minimal additional information. Furthermore, the solution is robust enough to provide usable results with as little as one distance measurement per image.

The proposed method requires a one-time extrinsic calibration for getting the transformation between the camera used for SfM reconstruction and the distance sensor. After that, the scale calculation can be done with only the captured distance measurements and an iterative method used to improve the accuracy of the computed scale, as well as the uncertainty brought about by the used distance sensor. To test the flexibility of the proposed method, three different distance sensors are used - from a low-cost single direction sensor to two 2D LiDAR solutions. The authors compare the proposed solution to two position-based scaling methods, using five different objects from different application scenarios. It is shown that the proposed method gives better performance to built-in scaling solutions and position based solutions, when used with comparably high-grade sensors and that the performance is still usable, when dropping to a low-cost single direction sensor.

Finally, it is shown that the scaling uncertainty is proportional to the input sensor's measurement uncertainty, and the proposed simplified method achieves comparable results to the more computationally intensive methods. This gives the possibility that both the scaling calculation approach and the uncertainty computation can be simply computed with only base sensor readings and modelling.

## References

- [1] H. Sarbolandi, D. Lefloch, and A. Kolb, "Kinect range sensing: Structured-light versus time-of-flight kinect," *Computer vision and image understanding*, vol. 139, pp. 1–20, 2015.

## References

- [2] M. Sarker, T. Ali, A. Abdelfatah, S. Yehia, and A. Elaksher, "a cost-effective method for crack detection and measurement on concrete surface," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, p. 237, 2017.
- [3] A. Corti, S. Giancola, G. Mainetti, and R. Sala, "A metrological characterization of the kinect v2 time-of-flight camera," *Robotics and Autonomous Systems*, vol. 75, pp. 584–594, 2016.
- [4] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer, "A survey of structure from motion\*." *Acta Numerica*, vol. 26, pp. 305–364, 2017.
- [5] I. Nikolov and C. Madsen, "Benchmarking close-range structure from motion 3d reconstruction software under varying capturing conditions," in *Euro-Mediterranean Conference*. Springer, 2016, pp. 15–26.
- [6] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, p. 78, 2017.
- [7] A. Martell, H. A. Lauterbach, A. Nuchtcer *et al.*, "Benchmarking structure from motion algorithms of urban environments with applications to reconnaissance in search and rescue scenarios," in *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2018, pp. 1–7.
- [8] K. Makantasis, A. Doulamis, N. Doulamis, and M. Ioannides, "In the wild image retrieval and clustering for 3d cultural heritage landmarks reconstruction," *Multimedia Tools and Applications*, vol. 75, no. 7, pp. 3593–3629, 2016.
- [9] C. W. VisualSFM, "A visual structure from motion system," 2011.
- [10] C. Sweeney, T. Hollerer, and M. Turk, "Theia: A fast and scalable structure-from-motion library," in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015, pp. 693–696.
- [11] J. L. Schonberger, "Colmap," <https://colmap.github.io>, 2016, accessed: 2018-09-06.
- [12] Agisoft, "Metashape (photoscan)," <http://www.agisoft.com/>, 2010, accessed: 2020-06-15.
- [13] Bentley, "Contextcapture," <https://www.bentley.com/en/products/brands/contextcapture>, 2016, accessed: 2020-06-15.
- [14] CapturingReality, "Reality capture," <https://www.capturingreality.com/>, 2016, accessed: 2020-06-15.

## References

- [15] R. Giubilato, S. Chiodini, M. Pertile, and S. Debei, "Scale correct monocular visual odometry using a lidar altimeter," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3694–3700.
- [16] D. Zhou, Y. Dai, and H. Li, "Reliable scale estimation and correction for monocular visual odometry," in *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2016, pp. 490–495.
- [17] L. Li and H. Lan, "Recovering absolute scale for structure from motion using the law of free fall," *Optics & Laser Technology*, vol. 112, pp. 514–523, 2019.
- [18] M. Rabah, M. Basiouny, E. Ghanem, and A. Elhadary, "Using rtk and vrs in direct geo-referencing of the uav imagery," *NRIAG Journal of Astronomy and Geophysics*, 2018.
- [19] D. Turner, A. Lucieer, and C. Watson, "An automated technique for generating georectified mosaics from ultra-high resolution unmanned aerial vehicle (uav) imagery, based on structure from motion (sfm) point clouds," *Remote sensing*, vol. 4, no. 5, pp. 1392–1410, 2012.
- [20] F. Clapuyt, V. Vanacker, and K. Van Oost, "Reproducibility of uav-based earth topography reconstructions based on structure-from-motion algorithms," *Geomorphology*, vol. 260, pp. 4–15, 2016.
- [21] I. Nikolov and C. Madsen, "Performance characterization of absolute scale computation for 3d structure from motion reconstruction," in *14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (Visigrapp 2019) International Conference on Computer Vision Theory and Applications*. SCITEPRESS Digital Library, 2019, pp. 884–891.
- [22] O. Kim and D.-J. Kang, "A sensor fusion method to solve the scale ambiguity of single image by combining imu," in *2015 15th International Conference on Control, Automation and Systems (ICCAS)*. IEEE, 2015, pp. 923–925.
- [23] T. Schöps, T. Sattler, C. Häne, and M. Pollefeys, "3d modeling on the go: Interactive 3d reconstruction of large-scale scenes on mobile devices," in *3D Vision (3DV), 2015 International Conference on*. IEEE, 2015, pp. 291–299.
- [24] L. Ding and G. Sharma, "Fusing structure from motion and lidar for dense accurate depth map estimation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 1283–1287.

## References

- [25] Y. Lu, J. Lee, S.-H. Yeh, H.-M. Cheng, B. Chen, and D. Song, "Sharing heterogeneous spatial knowledge: Map fusion between asynchronous monocular vision and lidar or other prior inputs," in *Robotics Research*. Springer, 2020, pp. 727–741.
- [26] A. Voulodimos, N. Doulamis, D. Fritsch, K. Makantasis, A. Doulamis, and M. Klein, "Four-dimensional reconstruction of cultural heritage sites based on photogrammetry and clustering," *Journal of Electronic Imaging*, vol. 26, no. 1, p. 011013, 2016.
- [27] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.
- [28] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [29] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European conference on computer vision*. Springer, 2006, pp. 430–443.
- [30] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.
- [31] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *International workshop on vision algorithms*. Springer, 1999, pp. 298–372.
- [32] Hokuyo, "Utm-30lx," <https://www.hokuyo-aut.jp>, 2010, accessed: 2020-03-06.
- [33] M. Daakir, M. Pierrot-Deseilligny, P. Bosser, F. Pichard, C. Thom, and Y. Rabot, "Study of lever-arm effect using embedded photogrammetry and on-board gps receiver on uav for metrological mapping purpose and proposal of a free ground measurements calibration procedure." *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2016.
- [34] D. Girardeau-Montaut, "Cloudcompare-open source project," *Open-Source Project*, 2011.
- [35] Slamtec, "Rplidar," <http://www.slamtec.com/en>, 2010, accessed: 2018-09-06.
- [36] M. A. Cooper, J. F. Raquet, and R. Patton, "Range information characterization of the hokuyo ust-20lx lidar sensor," in *Photonics*, vol. 5, no. 2. Multidisciplinary Digital Publishing Institute, 2018, p. 12.

## References

- [37] R. M. Haralick, "Propagating covariance in computer vision," in *Performance Characterization in Computer Vision*. Springer, 2000, pp. 95–114.
- [38] C. B. Madsen, "A comparative study of the robustness of two pose estimation techniques," *Machine Vision and Applications*, vol. 9, no. 5-6, pp. 291–303, 1997.
- [39] M. H. DeGroot and M. J. Schervish, *Probability and statistics*. Pearson Education, 2012.
- [40] M. Marčíš, "Quality of 3d models generated by sfm technology," *Slovak Journal of Civil Engineering*, vol. 21, no. 4, pp. 13–24, 2013.
- [41] Dawson-Haggerty, "trimesh," <https://trimsh.org>, 2019, accessed: 2020-03-06.
- [42] L. Wang and Z. Zhang, "Automatic detection of wind turbine blade surface cracks based on uav-taken images," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 9, pp. 7293–7303, 2017.
- [43] D. Zhang, K. Burnham, L. Mcdonald, C. Macleod, G. Dobie, R. Summan, and G. Pierce, "Remote inspection of wind turbine blades using uav with photogrammetry payload," in *56th Annual British Conference of Non-Destructive Testing-NDT 2017*, 2017.
- [44] J. Schöning and G. Heidemann, "Taxonomy of 3d sensors," *Argos*, vol. 3, no. P100, pp. 9–10, 2016.

## References

## **Part III**

# **Structure from Motion Data Analysis**



# Paper G

## Rough or Noisy? Metrics for Noise Estimation in SfM Reconstructions

Ivan Nikolov, Claus Madsen

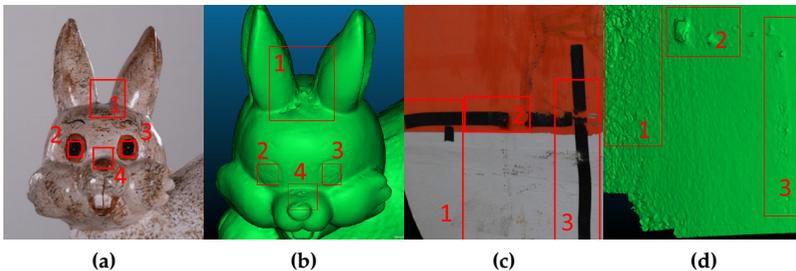
The paper has been accepted to  
*MDPI Sensors Journal*

*The layout has been revised.*

## Abstract

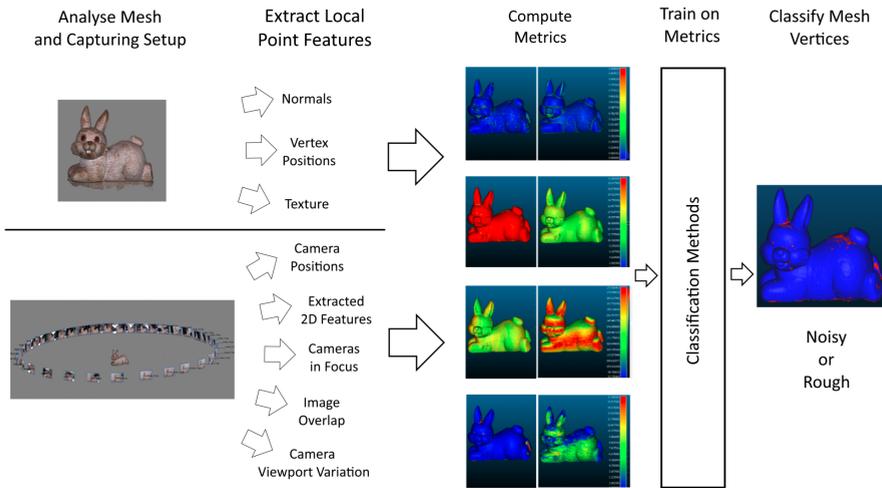
*Structure from Motion (SfM) can produce highly detailed 3D reconstructions, but distinguishing real surface roughness from reconstruction noise and geometric inaccuracies has always been a difficult problem to solve. Existing SfM commercial solutions achieve noise removal by a combination of aggressive global smoothing and the reconstructed texture for smaller details, which is a subpar solution when the results are used for surface inspection. Other noise estimation and removal algorithms do not take advantage of all the additional data connected with SfM. We propose a number of geometrical and statistical metrics for noise assessment, based on both the reconstructed object and the capturing camera setup. We test the correlation of each of the metrics to the presence of noise on reconstructed surfaces and demonstrate that classical supervised learning methods, trained with these metrics can be used to distinguish between noise and roughness with an accuracy above 85%, with additional 5-6% performance coming from the capturing setup metrics. Our proposed solution can easily be integrated into existing SfM workflows as it does not require more image data or additional sensors. Finally, as part of the testing we create an image dataset for SfM from a number of objects with varying shapes and sizes, which are available online together with ground truth noise annotations - <http://dx.doi.org/10.17632/xtv5y29xvz.2>.*

## G.1 Introduction



**Fig. G.1:** Illustration of SfM reconstruction geometrical errors, which need to be distinguished from real surface roughness. Noise parts are shown in red.

Structure from Motion is widely used for visualization and inspection purposes in the building [1–3], manufacturing [4] and energy industries [5], as well as for geology [6–8] and cultural preservation [9–11]. Because of the reliance of SfM on 2D image data, it is prone to geometric noise and topological defects, if optimal image capturing conditions are not met (Figure G.1).



**Fig. G.2:** Overview of the proposed idea for using metrics extract from the mesh and capturing setup used for SfM reconstruction, to determine if the underlying surface is noisy or rough.

This has prompted a number of benchmarks [12–14] on the accuracy and robustness of SfM solutions, as well as on the best possible lighting conditions, camera positions, image density and captured object surface characteristics. The problem of determining if noise is present on a 3D reconstructed mesh and differentiating between noise and the inherent roughness that surfaces and objects have is not a trivial one. Because topological defects and noise on the surface of SfM reconstruction are caused by a combination of sub-optimal capturing conditions, the surface properties of the scanned object and the camera used to capture the 2D, they cannot easily be quantified.

The main contribution of this paper is the exploration, development and evaluation of a number of metrics for determining if the underlying 3D reconstructed surface is noisy or rough. An overview of the idea proposed in this paper is shown in Figure G.2. The proposed metrics are chosen based on the known weaknesses of SfM solutions, as well as on the underlying principals used in many of the state of the art mesh simplification, quality assessment and denoising algorithms, given in the next section.

## G.2 State of the Art

Most of the commercial SfM solutions rely on global or isotropic smoothing algorithms. These algorithms remove noise, but smooth out smaller details. Reconstruction solutions like Metashape [15], ContextCapture [16], Reality

Capture [17], etc. use this approach, with additional options for mesh surface refinement. Such global denoising algorithms are also presented by [18–20].

Local feature or anisotropic algorithms analyse the underlying mesh geometry and normals to distinguish noisy areas from high surface roughness areas and preserve smaller details. The research from [21] uses a pre-filtering step and a  $L_1$ -median normal filtering, while [22] uses filtered facet normal descriptors and training of a neural network for calculating regression functions. Other research is focused on classifying normal regions and using isotropic neighbourhoods [23] or iterative estimation of normals and vertex movement [24, 25].

Another important factor for detecting noise is the geometric visibility of roughness, especially on complex surfaces. There are multiple proposed solutions by [26–29], using local visibility features, curvature calculation and normals to detect parts of meshes with low or high roughness. These methods are used both for detecting noise on a smooth meshes, but also for introducing watermarking to meshes without distorting their appearance.

Most of the described mesh denoising algorithms are not focused directly on SfM reconstructions and thus they do not use a lot of the information which can be taken from SfM production pipelines. In this paper we propose noise detection metrics, which can be used to distinguish noise caused by sub-optimal SfM reconstructions from the inherent roughness of the reconstructed objects. These metrics combine knowledge taken directly from 3D meshes reconstructed using SfM, with information taken from their textures, as well as from the camera setup used to capture the images used for reconstructing the object, such as camera positions, orientations, focal length and internal parameters. No external sensors or additional captured data are required for any of the presented metrics. With this our main contributions in this paper can be summarized:

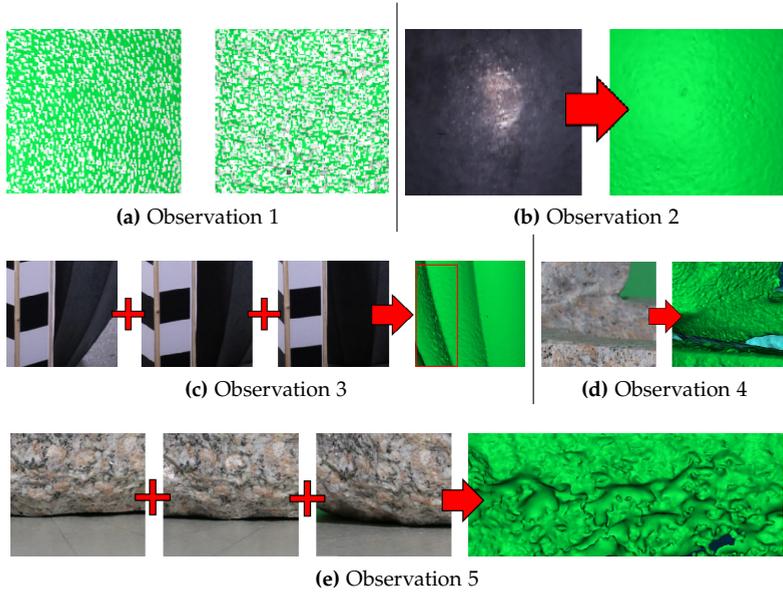
- We present a number of metric that can be easily calculated as part of the normal SfM workflow;
- We explore the correlation between each metric and the presence of noise on reconstructed objects;
- We train classical supervised learning methods using combinations of these metrics and demonstrate verify their accuracy;
- We test the metrics on a number of objects with varying surface textures, shapes and sizes to verify their robustness;
- We provide the captured database of images used to create the SfM reconstructions, together with the manually annotated ground truth data as part of the paper. This way others can use it for comparison and testing noise detection and removal implementations.

### G.3 Methodology

As part of this paper we propose nine metrics for detecting noise on SfM reconstructed meshes. These can be divided into two groups - metrics based on findings in the areas of mesh visual quality and roughness detection, and ones based on the SfM reconstruction weaknesses to sub-optimal capturing conditions. Five main observational hypotheses are made for the appearance of noise and geometric inaccuracies in SfM reconstructions and for each, one or more metrics are chosen as a way to describe each one. The observations are given in the numbered list below, with corresponding metrics shown in Table G.1. In the next sections, each of the metrics will be explained in detail. A visualization of each of the metrics on the surface of a reconstructed mesh is given in Figure G.4.

1. Noise manifests as either clumped together high frequency vertices or flat patches and holes - when the initial feature detection and matching methods in the SfM pipeline do not produce enough correct matches, the produced 3D surfaces can end up overlapping or missing parts. These manifest in geometrical surface errors, as seen in Figure G.3a;
2. SfM noise normally comes from smooth, monochrome colored surfaces - monochrome surfaces normally lack robust features like edges and angles, while smooth and transparent surfaces, produce reflections, which change with the view direction, making correct feature matching impossible (Figure G.3b);
3. Noise is present on parts of the object that have not been seen from enough camera positions - SfM needs to gather information of the object from multiple directions, to provide a correct geometrical representation of the micro and macro shape of the surfaces. Not enough camera variation can lead to 3D surface "guessing" and deformed patches. Example of this can be seen in Figure G.3c, where one object obscures another surface from being seen by the cameras resulting in noise;
4. Noise is present on parts of the object that have been seen from enough camera positions, but were not in focus - surface features need to be extracted and matched, but if parts of the object are blurred and out of focus, not enough information can be extracted from them. This is visualized in Figure G.3d, where the back of the object becomes out of focus, resulting in not enough features captured;
5. Noise is present on parts of the object that have been seen from enough camera positions, but those positions were not diverse enough - if all the capturing positions are from the same direction, not enough information can be extracted for the shape of the surface. This can be seen in

Figure G.3e, where multiple images are taken from a surface, but none of them have enough angular diversity in vertical direction, resulting in the reconstruction of the bottom of the surface being noisy.

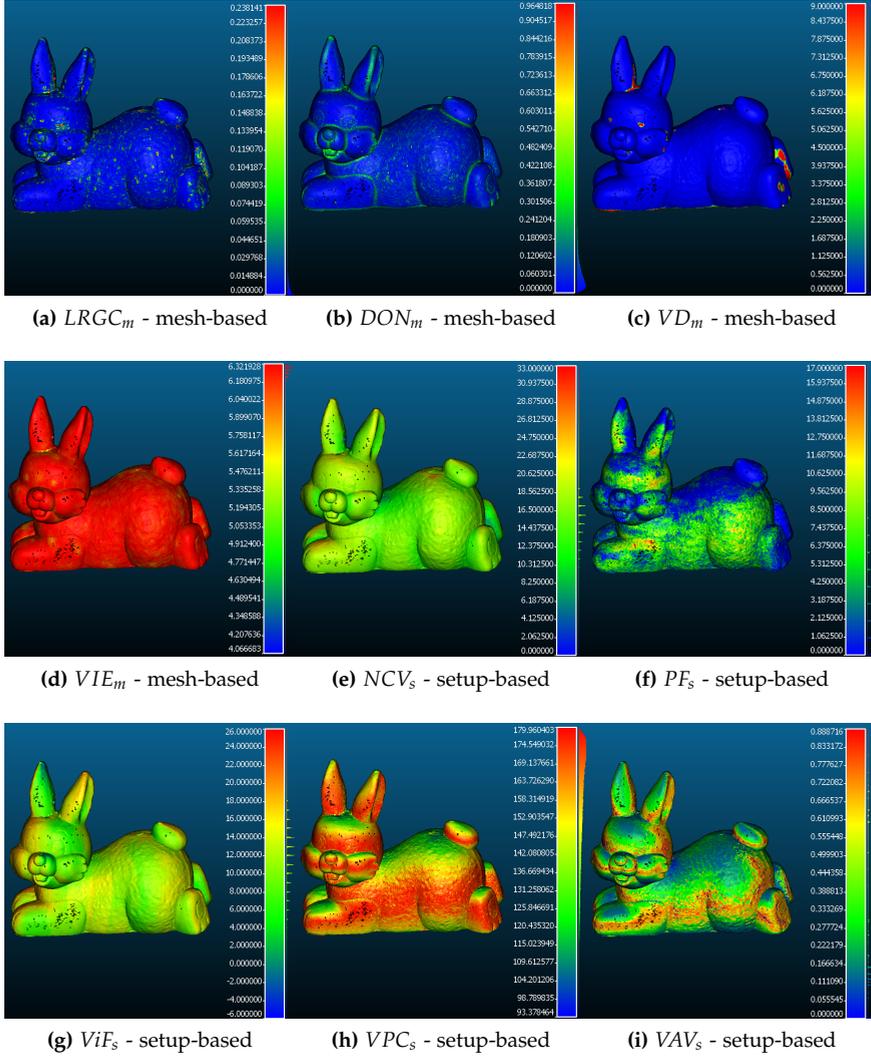


**Fig. G.3:** Examples of the five main observational hypotheses, used as a basis for the chosen mesh-based and capturing setup-based metrics

In the subsections below we will focus on each of the metrics' theoretical basis, extraction methods, interpretation, etc. For easier readability each of the metric abbreviations will have a subscript of  $m$  for mesh-based or  $s$  for capturing setup-based. Before computing each metric, the reconstructed object is scaled to absolute real-world scale. Once all the metrics have been presented, they will be analysed to determine their level of correlation. This will be presented in the Results section G.5.

### G.3.1 General Mesh-based Metrics

In this subsection we will cover the metrics extracted directly from the 3D reconstructed mesh. They are based on the vertex positions, normals and vertex color. These metrics are based on observational hypotheses 1 and 2, presented in Section G.3.



**Fig. G.4:** Visualization of all the proposed metrics as heat maps. For  $LRGC$ ,  $DON_m$ ,  $VD_m$ , higher values (indicated with red color) indicate higher risk of noise, while for  $VIE_m$ ,  $NCV_s$ ,  $PF_s$ ,  $ViF_s$ ,  $VPC_s$  and  $VAV_s$  - higher values, indicate lower risk of noise.

### G.3. Methodology

**Table G.1:** The five observational hypotheses and the chosen metrics, used to describe them. The different metrics are either based only on the reconstructed mesh itself or on the capturing setup - camera positions, intrinsic parameters, etc.

Num	Metrics	Type
1	Local Roughness from Gaussian Curvature ( $LRGC_m$ ) Difference of Normals ( $DON_m$ ) Vertex Local Spatial Density ( $VD_m$ )	Mesh-based
2	Vertex Local Intensity Entropy ( $VIE_m$ )	Mesh-based
3	Number of Cameras Seeing Each Vertex ( $NCV_s$ ) Projected 2D Features ( $PF_s$ )	Setup-based
4	Vertices in Focus ( $ViF_s$ )	Setup-based
5	Vertices Seen from Parallel Cameras ( $VPC_s$ ) Vertex Area of Visibility ( $VAV_s$ )	Setup-based

#### Local Roughness from Gaussian Curvature ( $LRGC_m$ )

**Rationale:** *Noise on the SfM surface appears as a geometric disturbance, which creates high roughness areas on otherwise smooth surface patches.*

The first calculated metric is the mesh’s local roughness, depending on a metric closely related to Gaussian curvature. The metric was first proposed by [27], in their paper for mesh quality assessment. Local curvature is widely used for visual quality assessment and denoising, as a characteristic describing the local changes of the surface. Their proposed algorithm first calculates the Gaussian curvature like metric (GC) in an area around each vertex, essentially describing how much the area deviates from a planar surface. This is done using Equation G.1, where  $N_i^{(F)}$  is all the neighbour faces around a point  $i$  and  $\alpha_j$  is the angle between the current vertex and the one which is incident to it.

$$GC_i = \left| 2\pi - \sum_{j \in N_i^{(F)}} \alpha_j \right| \quad (G.1)$$

Once the local curvature is calculated, a Laplacian matrix of the angles between the connected neighbours and each vertex is derived. Finally the local roughness metric LGRC is defined as a weighted difference between the Gaussian curvatures of each vertex and its neighbours, weighted according to the calculated Laplacian matrix. This is shown in Equation G.2, where  $D_{ij}$  is the Laplacian matrix and  $N_i^{(V)}$  is all the vertices in the neighbourhood of the current one. An in-depth explanation of the method can be seen in [27].

$$LRGC_i = \left| GC_i - \frac{\sum_{j \in N_i^{(v)}} (D_{ij} \cdot GC_j)}{\sum_{j \in N_i^{(v)}} D_{ij}} \right| \quad (G.2)$$

This metric is robust to curved surfaces and gives gradual and smooth values. The method gives a scale independent surface roughness measure. An example of the metric can be seen in Figure G.4a, where higher values denote higher roughness and higher risk of noise.

### Difference of Normals (DON<sub>m</sub>)

**Rationale:** *Noise on SfM surfaces appears as high frequency surface changes, especially on the edges of the mesh and surrounding holes in it.*

The metric is proposed by [30] and is used for surface roughness detection, point cloud segmentation, obstacle detection, etc. It is a scale dependent local value, sensitive to specific resolutions of roughness. Two radii  $r_1$  and  $r_2$  of different sizes are chosen around each vertex. The normals of the area below the neighbourhood for each radius are computed and their difference gives the final metric. Equation G.3 is used for calculating the difference of normals, where  $\hat{n}(p, r)$  is the normal of the surface under each of the radii for every vertex  $i$  and  $r_1 < r_2$ . Get the final measure, the magnitude of this vector is calculated, which is between  $[0, 1]$ .

$$DON_i = \left| \frac{\hat{n}(p_i, r_1) - \hat{n}(p_i, r_2)}{2} \right| \quad (G.3)$$

In their work, [30] demonstrate that high frequency areas contain smaller details in point clouds. SfM noise is normally represented as high frequency signal in clustered areas on the surface of the reconstruction. This is why we focus on capturing very high frequency surface changes. The radii are set heuristically to a percentage of the size of the reconstructed object - 2% of the size of the object, for the larger one and a factor of 10 smaller for the smaller radius, as suggested in [30]. This makes it independent from the scale of the object. With these input parameters, the difference of normals is especially sensitive to roughness at the edges of objects and allows it to provide a more focused additional roughness metric to  $LRGC_m$  metric. The calculated metric is visualized in Figure G.4b, where higher values denote higher difference between the local normals and higher risk of noise.

### Vertex Local Spatial Density (VD<sub>m</sub>)

**Rationale:** *When surface errors occur in SfM reconstructions, the resultant reconstruction contains areas of high vertex density, even on supposedly smooth real world object areas.*

This metric is based on point cloud segmentation methods like the one proposed by [31], using area of interest spatial neighbourhood grouping like K nearest neighbours. This metric is calculated by first computing a number of progressively larger search radii, connected to the overall size of the reconstructed object. The size is chosen heuristically and is in the interval  $R_{VD} = [0.1\% : 0.5\%]$  from the size of the object, as this is seen as the vertex density that best explains the possibility of noise. The mesh global maximum of neighbours for each of the radii is calculated. A percentage of these maximum values is taken and used as a threshold in the subsequent calculations. The lower this percentage is the less the local spatial density can be before it is viewed as problematic. For this paper the percentage is set to 60%.

For each vertex the number of neighbours is captured for each of the radii. If the number is above the threshold, a score is given for that vertex. The more instances get a number higher than the threshold, the higher the final score for that vertex. This is shown in Equation G.4, where  $N_i^{(r_j)}$  is the set of all neighbours for the current radius,  $N_{max}^{(r_j)}$  is the maximum set of all neighbours,  $DC$  is the density coefficient in percentage and  $s$  is the score. This way a vertex density score scaled to the global density of the object on multiple size levels is achieved. This makes the metric invariant to the scale of the object and it can be comparable between objects of different sizes. The calculated density metric is shown in Figure G.4c, where higher values indicate parts of higher vertex density and higher risk of noise.

$$VD_i = \sum_{r_j \in R_{VD}} s(j) \quad , \quad \text{for } s(j) = \begin{cases} 1, & \text{if } N_i^{(r_j)} \geq DC \cdot N_{max}^{(r_j)} \\ 0, & \text{otherwise} \end{cases} \quad (G.4)$$

### Vertex Local Intensity Entropy (VIE<sub>m</sub>)

**Rationale:** *SfM reconstruction tends to produce errors and noise when the object surface is featureless and monochrome [32].*

The intensity for each vertex is calculated from the texture RGB data. These intensities are then used to calculate the local entropy of the mesh. Color has been used for mesh and depth map denoising [33], [25] and it is shown to give good results. We choose to use entropy [34], as it can be more easily calculated locally on a point cloud, compared to other edge detection algorithms and can give a measure of the surface color intensity change. To calculate the entropy  $H$  we use Equation G.5, where  $P_i$  is probability of the occurrence of the specific intensity level at vertex  $p_i$  and  $N$  is the maximum number of possible intensity values equal to 256. The visualization of the entropy is given in Figure G.4d, where higher values indicate higher entropy and more varied surface color, with lower risk of noise.

$$H = - \sum_{i=0}^N P_i \log_2 P_i \quad (\text{G.5})$$

### G.3.2 Capturing Setup-based Metrics

The following metrics are unique for SfM meshes, as they are extracted from the camera capturing setup and utilize the position, orientation, view density of the cameras, etc. The main factors for selecting these metrics, are the dependencies demonstrated by [14, 35, 36], between the quality of the capturing setup and the resultant reconstruction. To calculate these metrics a Unity implementation is created for positioning the reconstruction and calculated camera positions, as well as reprojecting the necessary data. We use the Unity engine, because of the easy programming pipeline using C#, fast ray cast computation and the possibility to visualize and compute large 3D model relatively fast and easy. An overview of the used development pipeline is given in Section G.4. These metrics are based on the hypothesis observations 3, 4 and 5.

#### Number of Cameras Seeing Each Vertex (NCV<sub>s</sub>)

**Rationale:** *To create a good SfM reconstruction, a high amount of overlap between images is required [9], [11], which means that vertices "seen" by many cameras have a lower risk to contain noise.*

To compute this metric, all the pixels of each of the calculated cameras are projected to the reconstructed mesh. The metric is calculated by projecting the captured images from the calculated camera positions towards the reconstructed mesh. Each vertex is scored depending on the amount of image pixels projected onto it, meaning that the higher the score the more cameras have "seen" the vertex. The visualization of the metric is shown in Figure G.4e.

This metric gives an overview of how certain we are, the data created by the SfM system is representative of the real world object. If not enough cameras see certain parts of the objects, there is a bigger chance that those parts will contain noise or holes. The following metrics will expand on the information captured by this metric.

#### Projected 2D Features (PF<sub>s</sub>)

**Rationale:** *To create the SfM reconstruction, 2D feature points are extracted from each image. These features are matched between images and used in the triangulation of the sparse point cloud and the reprojection of camera positions [37]. By projecting these points to the mesh, areas of higher certainty can be found, by exploiting the fact*



**Fig. G.5:** An image used as input to the SfM solution and calculated feature points. A radius is set around each of the features and all points that are in the area are projected to the reconstructed mesh

*that areas not containing any found and matched features, will produce lower quality reconstructions*

We look at the 2D features extracted in the triangulation and camera position calculation step of the SfM pipeline. In this step features are extracted from each image and matched between them. In most SfM solutions, these 2D feature descriptors are not disclosed, but they are mostly variations of SURF [38] or free alternatives like FAST [39] and ORB [40]. An example image with captured feature points can be seen in Figure G.5, where it can be seen that smooth areas like the eyes and noise of the bunny statue have much less features. For each camera position, the already calculated feature descriptor points are extracted. A radius around each point is set and the points under that area are projected to the 3D reconstructed model. For each 3D point the metric as aggregated depending on how many of these matched feature point areas are projected onto it.

The higher the value of this metric for each vertex, the more 2D features were projected onto it. Figure G.4f shows this metric. As these 2D features are used in the reconstruction itself it is hypothesized that a high metric will have less noise.

#### **Vertices in Focus (ViF<sub>s</sub>)**

**Rationale:** *Structure from Motion matches points between images for creating the initial sparse point cloud and camera position and orientation calculation. If parts of the object are captured out of focus, these points would have blurring on them. This can increase the possibility for reconstruction noise to be present in these parts.*

To calculate the metric, first the near  $N_p$  and far  $F_p$  focal plains are calculated for each camera using the formulas presented in Equation G.6. There  $H_f$  is the hyperfocal distance, which is the distance between the camera and the closest surface, which is in focus, when the lens is focused on infinity, while the  $CoC$  is the circle of confusion calculated according to [41]. The focal length  $F$  and aperture  $A$  are known from the EXIF data present in the images and the distance to the object  $D$  is calculated from the camera to the closest surface of the reconstruction. Because the object is scaled before capturing the metrics, the measured distances between cameras and the object should be in correct units.

$$N_p = \frac{H_f \cdot D}{H_f + (D - F)}, \quad F_p = \frac{H_f \cdot D}{H_f - (D - F)} \quad (G.6a)$$

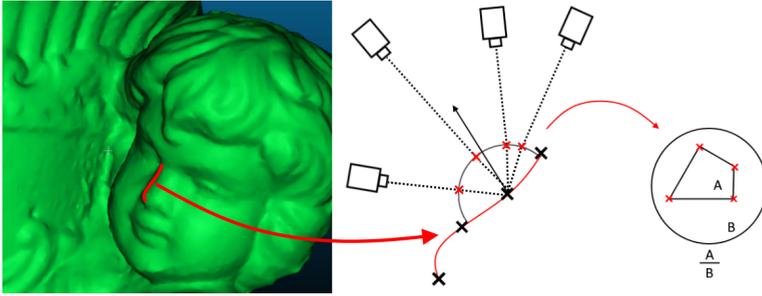
$$H_f = \frac{F^2}{ACoC}, \quad CoC = \frac{F}{1720} \quad (G.6b)$$

A ray is cast from each pixel of the camera, to the corresponding face from the reconstructed model and the distance between the two is calculated. Vertices of faces outside of the focal planes are scored with -1 for cameras which have seen them, while ones that are inside the focal planes are scored with 1. A lower score indicates more out of focus cameras having seen the vertex and a higher chance of it being noisy. The metric can be seen in Figure G.4g, where the lower the value, the more times it has been out of focus and the higher risk for noise.

### Vertices Seen from Parallel Cameras (VPC<sub>s</sub>)

**Rationale:** *Even if multiple images have captured the surface of the object, if all of them "see" it from large angles, without at least one central image to connect them, there is a possibility of SfM calculation error [42].*

This metric is captured by computing the angle between each normal and the forward direction of each of the calculated cameras that can "see" the vertex. This is achieved by using Equation G.7, where  $\alpha_m$  is the calculated angle between the normal  $N_i$  of vertex  $v_i$  and the camera forward direction vector  $C_f$  for each camera seeing the vertex  $[0, i]$ . Two 3D vectors are parallel, if the angle between them is either 180 or 0 degrees, but the camera has to be able to see the vertex, so an angle of 0 degrees is not likely. The closer at least one angle is to 180 degrees, the less chance there is of noise. Figure G.4h



**Fig. G.6:** Visualization of the calculated hemisphere positioned above each vertex in the mesh and the camera position, together with the intersection points. The distance from the camera to the vertex position is made in a smaller scale for easier visualization. Once all the intersection points are found the area between them is calculated and the ratio between it and the whole area is used for the metric

show this metric.

$$\alpha_i = \arccos \frac{C_f \cdot N_i}{|C_f \cdot N_i|} \quad (\text{G.7a})$$

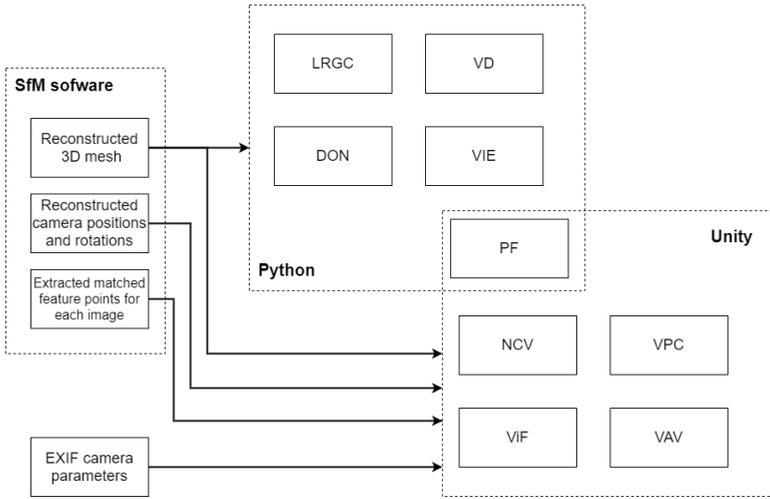
$$\alpha_{max} = \max_{\{1:i\}} \alpha_i \quad (\text{G.7b})$$

### Vertex Area of Visibility (VAV<sub>s</sub>)

**Rationale:** *To capture a surface's shape, SfM requires images from multiple positions and angles, so all parts of the topology are visible. If only little variation is given in the imaging positions, the resultant mesh can exhibit noise patches, surface deformations and holes [42].*

The metric requires the calculation of the area in space, from which each vertex is seen. We assume that the object surface is visible from every camera point of view. To model this metric, first a hemisphere is placed on the position of each vertex, oriented depending on the underlying normal. A hemisphere is chosen, as the assumption is that the cameras need to be able to physically see surface and the presence of self-occlusion. A ray is cast from each camera that "sees" the vertex. The points of intersection between each ray and the hemisphere are calculated and their 3D coordinates are saved. An example of this can be seen in Figure G.6, with the camera position pulled closer and the hemisphere colored for easier visualization.

We then project the points in 2D, to avoid working with spherical geometry. The Lambert azimuthal equal-area projection, is chosen as it represents correctly the area in all regions of the sphere. For the projection Equation



**Fig. G.7:** Overview of the implementation pipeline, showing what input and programming environments are used to calculate each of the metrics. The mesh-based metrics are directly computed in Python, while the capturing-setup based ones use a combination between Python and the Unity game engine.

G.8 is used, where  $(x, y, z)$  are the Cartesian coordinates of the points on the sphere and  $(X, Y)$  are the projected ones. The metric is calculated as a ratio between the area of the projected points and the whole area. An example of the metric can be seen in Figure G.4i, where the higher the values are, the higher the area of visibility is and the lower the risk of noise. This means that even if a lot of cameras have seen the point if their angular coverage from different positions is not large enough this would be penalized.

$$X = \sqrt{\frac{2}{1-z}}x, \quad Y = \sqrt{\frac{2}{1-z}}y \quad (\text{G.8})$$

## G.4 Implementation

In this section a short overview of the implementation pipeline is given. The different processing environments for extracting each of the metrics are given in Figure G.7. The initial data of the reconstructed mesh, the camera positions and orientations and extracted feature points are taken directly from the SfM software. For our current implementation Agisoft Metashape [15] is used, but the same data can be extracted from many of the commercial and open source SfM applications. In our case Metashape uses a Python based API for automation of the SfM pipeline, which can be also used to extract the

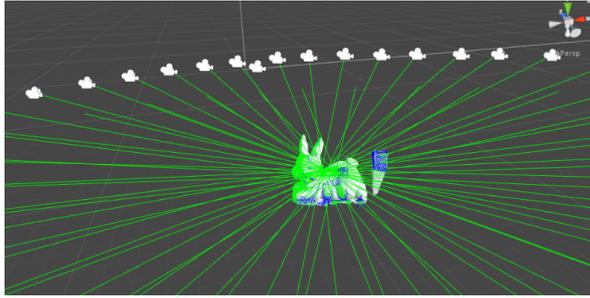
required data and parse it in a structure, used for metric extraction. For the purely mesh-based metrics only the reconstruction itself is used and the processing is done directly in Python. For extracting data and manipulating the 3D data, the library open3D [43] is used in. The extracted features are manipulated and the areas around them calculated, by using OpenCV [44] for Python. The capturing setup-based metrics are calculated through the use of the Unity game engine [45]. The engine uses C#, with specific optimizations for vector and GPU computations. Normally used for making games and interactive experiences, we use the powerful 3D features of the engine, the camera settings and the fast and easy ray calculating capabilities. The data from the Metashape Python API in these cases is saved to a custom format containing all the mesh data - vertices, faces, normals, color information, as well as camera positions and orientation. For these metrics, the EXIF data from each image is also used, for calculating the proper field of view and depth of field of each of the cameras. The setup-based metrics are calculated per mesh vertex, by casting rays from each pixel of the camera positions to the reconstructed surfaces. An example view of the Unity implementation is given in Figure G.8a, where the reconstruction together with the calculated camera positions and their forward direction vectors are given. The projected points on the mesh are used to calculate the NCV metric and show which parts of the object are seen by the particular camera. The input photo and the equivalent view from the Unity camera are given in Figure G.8b and G.8c

## G.5 Testing and Results

Testing the proposed metrics is done in a number of steps. First the correlation between the different metrics is calculated. This will give an initial idea if any of them give redundant information, too similar to the others. The second step is to create a dataset of images and SfM reconstructions. These objects have varied sizes, shapes, roughness levels and are made from different materials with different textures. We then manually annotate each one of the reconstructions on a vertex level - as noise and not noise. This annotation is used as ground truth for testing the accuracy of the proposed metrics.

We then separate the reconstructed objects into testing and training data and use the metrics together with the annotated data to train a number of supervised learning classification methods. The accuracy of the proposed metrics can then be evaluated for segmentation of the testing data into noise and not noise vertices.

To evaluate if all metrics are useful for detecting noise, we first calculate the correlation between the appearance of noise and each of the metrics. We then use that information to retrain the best performing supervised classification method on different subsets of the metrics and evaluate the resultant



(a) View from NCV metric calculation



(b) Input image

(c) Unity camera view

**Fig. G.8:** Views from the Unity implementation used for the capturing setup-based metric extraction.

accuracy.

Finally, we also evaluate the proposed solution in a wider industrially relevant context, by using a reconstruction of a wind turbine blade for testing and evaluating the results from it.

### G.5.1 Data Gathering

To ensure the robustness of the proposed metrics, objects with different shape, size, roughness and color, as well as material are used. All the objects are shown in Figure G.9. Special care was taken to create a diverse set of objects, to lower the possibility of bias in the proposed metrics. Some of the ways the dataset can be separated:

1. By size of the objects - we have objects ranging from 150 mm (cups shown in Figure G.9i, G.9j, etc.) to 800 mm (the black vase Figure G.9d and sea vase Figure G.9f), together with the wind turbine blade segment (Figure G.14a), which is more than 1500 mm long;
2. By material - we have objects made from stone, ceramics, plastic, clay, wood and metal. This guarantees that we can have varying surface properties like reflectivity, texture and color variation;

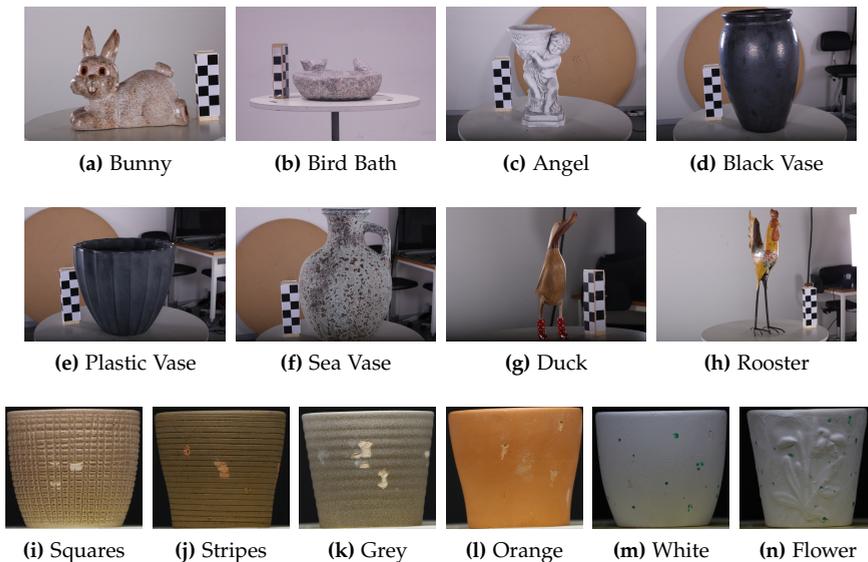
3. By shape complexity - we have objects with simple shapes and repeated patterns like the different cups and vases, as well as objects complex shapes. With all the possible problems that can arise from that - self-occlusion (Figure G.9c) or thin and narrow regions (Figure G.9g, G.9h).

A Canon 5Ds DSLR camera is used for capturing images of the objects. The resolution was set to 8688x5792 and a zoom lens with a variable focal length of 30-105 mm was used. The zoom lens was used, so the focal length can be easily changed depending on the size of the object. The focal length was set at the start of the capturing process for each object and kept the same throughout, only being changed if needed, once a new object is selected. This is done to guarantee, that the captured object is always in frame and most parts of it also in focus. The focal length was changed depending on the size of the object. For the initial and subset tests 36 images were taken in a circle around each object in one horizontal band. The camera is setup to such a height, so it stays perpendicular to the side of the objects. The research by [14], shows that this one vertical band capturing setup ensures that the objects can be reconstructed, but there is a possibility of geometrical noise on their surfaces. For the industrial context test 2x17 images in vertically stacked horizontal bands were used, because of the larger size of the wind turbine blade, compared to the objects used in the initial and subset. This way the front of the blade can be captured and reconstructed. All the objects were reconstructed using Agisoft Metashape and all the required data - camera positions, orientations, internal parameters, etc. was extracted from the program workflow, as explained in Section G.4. To make them more manageable to work with the reconstructions are sub-sampled to around 50k vertices. The actual number depends on the size and complexity of the shape of the object.

For testing the proposed solution and training the classification methods, a roughness/noise ground truth is created for all the used objects. The ground truth is made manually by annotating all the reconstructed meshes and masking all vertices of surfaces containing noise or any other topological defects (Figure G.12). The software used for annotation of the mesh vertices is also developed in Unity (Figure G.10) and at the end of the process the information for each vertex for each of the objects is saved into an array of values - showing 0 for clear surfaces and 1 for noise and geometrical defects. This annotated data is also used for testing the correlation between the appearance of noise and the different metrics.

### G.5.2 Correlation Analysis

The correlation between the different independent metrics needs to be tested, to ensure that highly correlated ones are removed, as they do not give any new information and can introduce uncertainty and interfere the detection of



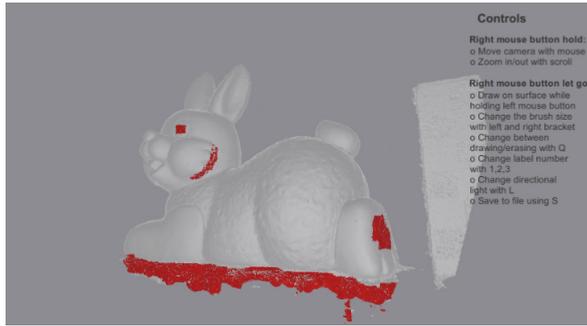
**Fig. G.9:** Objects selected for the robustness test. These objects have widely varying shape, size, roughness profiles and materials.

the noise. In addition, the correlation between the metrics and the appearance of noise is also analysed. To compute the correlation between the metrics a correlation matrix is calculated using the Pearson correlation coefficient [46]. The matrix is shown in Figure G.11.

We choose to consider a cutoff between metric correlation higher than 0.5 and with the dependent variable lower than 0.1. From the correlation matrix it can be seen that one of the metrics has a high correlation with the others - the number of cameras seeing each vertex ( $NCV_s$ ). Because this metric is quite generic and much of the information that it carries is present in the vertices seen from parallel camera ( $VPC_s$ ), with correlation of 0.65 and the vertex area visibility ( $VAV_s$ ), with correlation of 0.53, as well as projected 2D features ( $PF_s$ ) metric, we choose not to include  $NCV_s$  in the final set of metrics.

The correlation between the independent variable metrics and the dependent variable, which in our case is the presence of noise and geometric inaccuracies, is further explored. From the correlation matrix in Figure G.11, we can deduce that three mesh roughness metrics  $LRGC_m$ ,  $DON_m$  and  $VD$  have the highest correlation with the presence of noise. This is expected as these metrics are directly connected to the topology of the mesh. From the capturing setup-based metrics the most correlated ones to the presence of

## G.5. Testing and Results



**Fig. G.10:** View from the annotation tool used for creating the roughness versus noise ground truth for each of the meshes. The vertices painted red are set as reconstruction noise

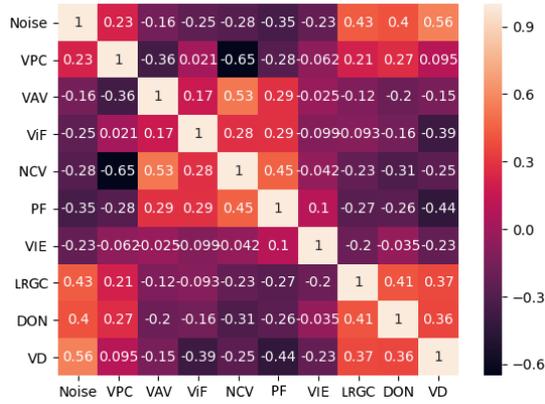
noise are  $PF_s$ ,  $NCV_s$ ,  $ViF_s$ , but  $NCV_s$  has been removed from consideration due to the high correlation with the other metrics. These observations will be used in Section G.5.4, when different subsets of the metrics are tested out.

### G.5.3 Initial Testing

For the initial test we use all the proposed metrics, except  $NCV_s$ . Further testing of subsets of metrics, will be given in Section G.5.4. The metrics are used to train a number of supervised learning classification methods - support vector machines (SVM), K-nearest neighbours (KNN), naive Bayes (NB), decision trees (DT), as well as more complex ensemble methods - random forests (RF) and AdaBoost (AB). The implementations are taken from Scikit-learn [47]. The hyperparameter used for each classifier are given in Table G.2. Because of the limited number of test objects, we use a cross validation, where we train on all, but one and test on it. We do this for each of the objects. Because the two classes - noise and not-noise are not balanced, an oversampling strategy is deployed when pre-processing the training data. The oversampling is done using Synthetic Minority Over-Sampling Technique (SMOTE) [48].

Because of the imbalanced dataset, we focus not only on the accuracy, but on the precision, recall and F1-score, which are shown in Table G.3. The table presents the average of all calculated performance factors for all the tested objects. From these, the AdaBoost classifier provided the best results, depending on the combination of the calculated factors.

All the tested classifiers give satisfactory results, with high recall, which indicates that it classifies noise vertices as such. On the other hand they also classify non-noise vertices as noise, which is shown by the low levels of precision. This shows that metrics can be useful for signalling to possible areas of noise and can be a part of a semi-automatic SfM noise detection pipeline, where a user then verifies the results. For a better understanding of the per-



**Fig. G.11:** Correlation matrix of the used metrics, together with the dependent variable. For easier visualization the metrics are shown with their coded names -  $VPC_s$ : vertices seen from parallel camera,  $VAV_s$ : vertex area visibility,  $VIF_s$ : vertices in focus,  $NCV_s$ : number of cameras seeing each vertex,  $PF_s$ : projected 2D features,  $VIE_m$ : vertex local color entropy,  $LRGC_m$ : local roughness from Gaussian curvature,  $DON_m$ : difference of normals and  $VD_m$ : vertex local spatial density.

**Table G.2:** Used hyperparameters for the tested classification methods - support vector machines (SVM), K-nearest neighbours (KNN), naive Bayes (NB), decision trees (DT), random forests (RF) and AdaBoost (AB).

Method	Parameters
SVM	$C = 8$ , kernel = linear, gamma = scale
RF	n_estimators=150, max_depth=10, min_sample_split = 3
AB	n_estimators=150, learning_rate = 0.5
KNN	n_neighbors = 5, weights = uniform, algorithm = auto
NB	default parameters
DT	criterion= entropy, max_depth=10, min_sample_split = 2

**Table G.3:** Average results from the fourteen objects and the chosen classical classifiers - support vector machines (SVM), K-nearest neighbours (KNN), naive Bayes (NB), decision trees (DT), random forests (RF) and AdaBoost (AB).

Method	$\overline{ACC}$	$\overline{Precision}$	$\overline{Recall}$	$\overline{F_1}$
SVM	0.816	0.569	0.842	0.679
RF	0.824	0.580	0.879	0.699
AB	0.851	0.630	0.844	0.742
KNN	0.812	0.568	0.789	0.660
NB	0.809	0.558	0.832	0.668
DT	0.824	0.578	0.885	0.699

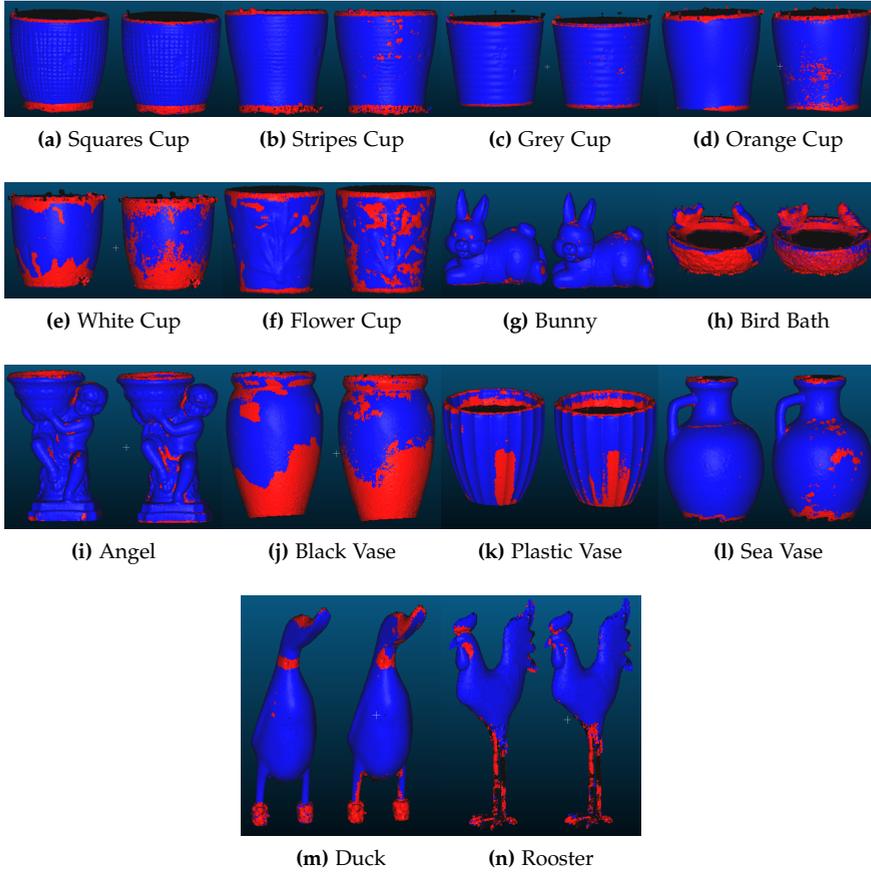
formance of the best performing model, the pseudo-colored visualizations of the annotated and classified noise vertices are also given in Figure G.12. Some notable problem areas can be seen on rough objects like the bird bath (Figure G.12h) and the sea vase (Figure G.12l) and objects with complex shapes, which have been reconstructed correctly, but have small noise patches - like the bunny (Figure G.12g) and the angel statue (Figure G.12i). On these objects the noise is either closely integrated into the surface roughness frequency or is quite sparse, compared to the overall size of the object.

Further complicating the non-trivial task are the manually annotated areas. For example, in the case of the two white cups (Figure G.12e and Figure G.12f) the overall low reconstruction accuracy, means that there is noise with different levels of severity. Where the cutoff between acceptable surface and noise is, can become very arbitrary, without classifying the whole surface as noisy. One way to alleviate this is to have multiple people annotate the same objects and get an average annotation. This will be further explored in the Conclusion and Future Work section G.6.

### G.5.4 Subset Testing

The calculated results in the previous section are based on all metrics except  $NCV_s$ . To test how much influence each of the metrics have on the calculated performance, a number of subset tests are performed. Five main tests are setup as shown in Table G.4. Because both the  $LRGC_m$  and  $DON_m$ , are used in the literature for point cloud classification, they are used separately, as a baseline, naive, first test for detecting noise on SfM reconstructions. The second test checks if  $NCV_s$  will have negative influence in the results, because of its high correlation with  $VPC_s$  and  $VAV_s$  metrics. All other metrics are used for this test scenario. Using the information gathered in Section G.5.2, the  $LRGC_m$ ,  $DON_m$  and  $VD_m$  are set as main metrics, because of their high correlation with the presence of noise. The third scenario tests how important are the mesh and capturing setup-based metrics for the performance of noise detection. The fourth test takes the three designated main metrics and creates five subsets, but adding each of the capturing setup-based metrics, to see how important are they separately. The final test again takes the main metrics and combines them with the other ones, which are either the more correlated or the less correlated to the noise.

The best performing classification method from the initial test is chosen for this scenario - AdaBoost. It is retrained with the different subsets of metrics and the results are given in Table G.5. Again the average of the calculated performance factors using the leave one out strategy for cross validation. For visualization purposes the resultant detected noise from each subset, for one of the test objects is shown in Figure G.13, together with the ground truth annotated noise.



**Fig. G.12:** The annotated ground truth vertices on the left and the same classified vertices using our proposed method on the right. The noise vertices are colored red, while the non-noise ones are blue.

**Table G.4:** Four main subset test scenarios. Each of the scenarios is designed to test the impact of different metrics or combination of metrics on the final results.

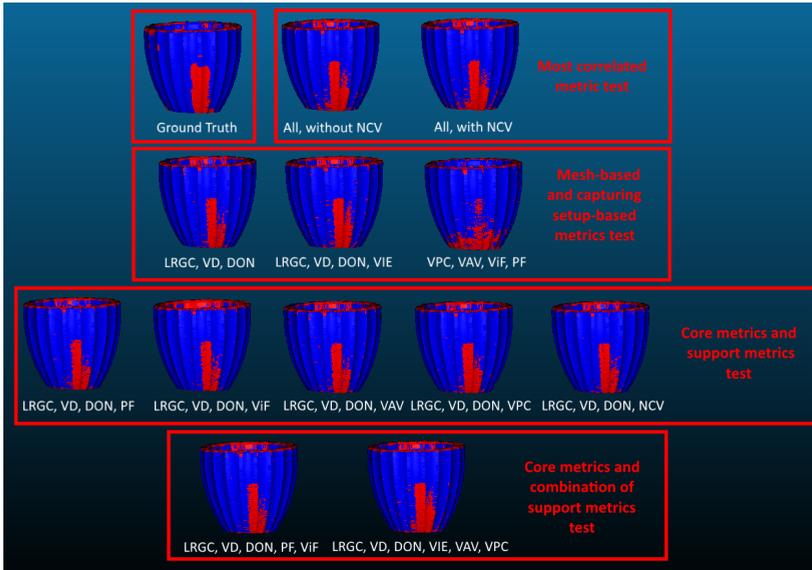
Test	Description
1	$LRGC_m$ and $DON_m$ separately
2	All metrics, with and without $NCV_s$
3	Mesh-based versus capturing setup-based metrics
4	Each capturing setup-based metric's impact
5	Impact from different combinations of setup-based metrics

## G.5. Testing and Results

**Table G.5:** Results from testing different subsets of the proposed metrics. Each of the subsets is used to train the best performing classification method from the first testing scenario AdaBoost. Different subsets are created to test the posed question in Table G.4

Subsets	ACC	Precision	Recall	F1	Test
Only $LRGC_m$	0.723	0.492	0.652	0.574	1
Only $DON_m$	0.686	0.407	0.788	0.537	1
All, without $NCV_s$	<b>0.889</b>	<b>0.674</b>	0.863	<b>0.756</b>	2
All, with $NCV_s$	0.852	0.635	0.848	0.725	2
$LRGC_m, VD_m, DON_m$ (Mesh-based)	0.828	0.592	0.833	0.692	3
(Mesh-based) + $VIE_m$	0.837	0.611	0.822	0.701	3
$VPC_s, VAV_s, ViF_s, PF_s$	0.707	0.425	0.753	0.544	3
(Mesh-based) + $PF_s$	0.840	0.615	0.829	0.706	4
(Mesh-based) + $ViF_s$	0.838	0.615	0.809	0.699	4
(Mesh-based) + $VAV_s$	0.837	0.612	0.811	0.698	4
(Mesh-based) + $VPC_s$	0.839	0.614	0.824	0.704	4
(Mesh-based) + $NCV_s$	0.831	0.603	0.799	0.701	4
(Mesh-based) + $PF_s, ViF_s$	0.814	0.565	<b>0.869</b>	0.683	5
(Mesh-based) + $VIE_m, VPC_s, VAV_s$	0.839	0.615	0.822	0.703	5

The naive approaches to using only the  $LRGC_m$  and  $DON_m$ , yield overall lower results, showing that only analysing the roughness profile of the reconstruction, cannot completely separate noise from real world surface roughness. The results also show that, as expected, the mesh-based metrics give the highest effect on the performance of the classification method, meaning that they are the most useful in discriminating between noise and surface roughness. The texture metric  $VIE_m$  helps boosting the overall accuracy and precision of the detection. This can be seen in Figure G.13, with a lot less random noise vertices, compared to the purely  $LRGC_m, VD_m, DON_m$  trained detector. The capturing setup-based metrics on their own are too vague to properly discern between noise and surface roughness, as seen from the lower overall accuracy. When introducing them to the mesh-based metrics, it can be seen that they also boost the overall performance when segmenting the noise from the roughness. Overall different combinations of the metrics can be useful in different situations, depending if it's more important to detect more of the noise correctly, but also miss-classify some of the roughness as noise, or vice-versa. The combination between the mesh-based metrics with the different capturing-setup metrics also shows that depending on the structure of the objects different capturing metrics can be useful. Larger objects benefit more from the  $ViF_s$  and  $VPC_s$  metrics, while smaller objects benefit more from  $VAV_s$  and  $VPC_s$  metrics. The  $PF_s$  metrics is the one that always gives positive impact to the performance, as it is directly connected to the captured 2D



**Fig. G.13:** Visualization of the noise detection results, using different subsets of metrics, together with the ground truth annotation. The different testing scenarios are separated for easier comparison.

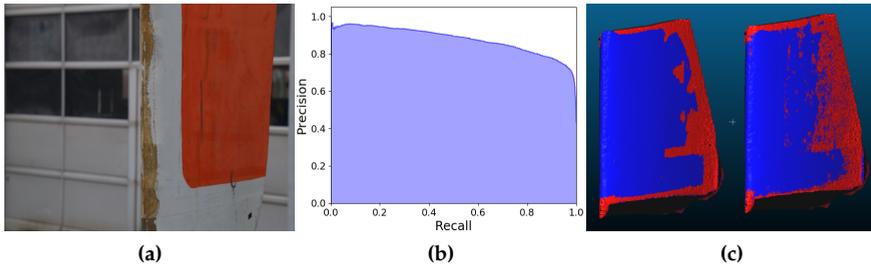
feature points.

### G.5.5 Industrial Context Test

The final test is made to give a wider industrial application context to the proposed metrics. We want to test if the described metrics can be used on data from different areas. This will also provide a better understanding on the generalization capabilities of the proposed metrics. We choose to test on wind turbine blade data, as this is an industrial inspection area, which has began to use SfM for capturing information more and more and research is focused on ensuring the high quality of the reconstructions [49]. In addition, wind turbine blade data is hard to acquire, because of the requirements by blade manufacturers, that blades in use are not normally imaged. If the proposed metrics can be used to train noise recognition methods on generic data and then can be used on wind turbine blade surface reconstructions, it would make researching and benchmarking SfM results from blades surfaces much more easily accessible.

For the test, a decommissioned wind turbine blade segment is selected (Figure G.14a). To ensure that the blade has different types surface roughness and damaged areas, it has been additionally sandblasted. The image

## G.5. Testing and Results



**Fig. G.14:** The wind turbine blade used for the second testing scenario (G.14a), together with the precision-recall curve of the classification model (G.14b) and the visualized annotation compared to classified vertices (G.14c). Red vertices are noise, blue are non-noise.

capture was done in an outdoor environment. Because the object is considerably larger than the ones used in the previous tests and normally the leading edge and sides of blades are inspected, a different image capture pattern is selected. Two vertical bands of 17 images in a semi-circle pattern are captured, leading to 34 images in total. The best performing classifier is chosen from the first two tests - AdaBoost.

We choose also the best performing combination of metrics - all, except  $NCV_s$ . All the reconstructions used in the previous testing scenarios are used as training data for AdaBoost. To evaluate the performance of the metrics on the blade, ground truth noise and roughness annotations are also made for it. The calculated classification results have an accuracy of 0.843, while the precision is 0.786 and recall is 0.877. For this test the precision-recall curve is also calculated for giving a better idea of the performance of the trained model using the proposed metrics (Figure G.14b). We chose to use it instead of a ROC curve, on the basis of the unbalanced dataset. This way the calculated results are going to be less skewed and "optimistic" [50]. The area under the curve (AUC) of the precision-recall curve is 0.877. Finally, the pseudo-colored visualization of the classified and annotated vertices for the wind turbine blade model are given in Figure G.14c. Overall the metrics provide acceptable results, by capturing all the problem areas around the top, bottom and back of the object, without miss-classifying the real damaged areas of the edge of the blade. This shows that a transfer learning effect can be used, where the training can be done on more easily accessible generic 3D reconstruction objects and how noise is seen on them and then the trained classifier can be used on specialized input data like wind turbine blades, with high level of accuracy.

## G.6 Conclusion and Future Work

The problem of detecting noise and geometric disturbances of 3D reconstructed meshes resulting from SfM is a non-trivial one. In these meshes noise and regular surface roughness can exhibit the same characteristics, making it difficult for detecting noise without miss classifying the roughness. This is why in this paper we present a number of metrics based on both the mesh surface and on the capturing setup. This combination of metrics is chosen, as it has been observed from the state of the art in SfM testing and benchmarking, that the appearance of geometrical errors and noise on the reconstructions is highly correlated to the quality of the capturing setup, the used camera and the number of images taken. By combining these metrics and analysing their performance we are trying to address a gap in the knowledge of SfM results and how they can be used in applications like industrial inspection and surface roughness estimation. In addition, none of the proposed metrics require external sensor data and can be easily integrated in normal SfM production pipeline.

To test the metrics a dataset of images is captured from a number of objects with different shapes, sizes, textures and materials. These objects are then reconstructed and the metrics are captured from them. The amount of correlation between the metrics and between the metrics and the presence of noise is computed and is seen that only one of the metrics - the  $NCV_5$  is highly correlated to the others. A number of classical supervised learning classification methods are trained on the metrics, together with ground truth manually annotated data. The results from classifying the meshes as noisy and not noisy vertices are shown to be usable, with the metrics generally giving a good overview which parts of the meshes contain noise, with some noise miss-classified as roughness. On the other hand surface patches, which contain real life damages are correctly classified as not noise. The captured dataset of images, together with the ground truth annotations will be available online for use for training and testing purposes.

Different combinations of the proposed metrics are also tested, to see how individual metrics influence the performance of detecting noise. We demonstrate that a naive approach of just using the roughness of the surface of the reconstruction does not yield high quality results, with an overall accuracy between 0.68 to 0.72. The results could be dramatically improved by introducing a combination of all the mesh-based metrics proposed in the paper, pushing the accuracy to 0.85. The mesh-based metrics manage to describe the rough parts of objects, but tend to be less discriminative between the parts with high roughness and the ones with geometrical errors. The use of capturing setup-based metrics is shown to be helpful in discerning between the two, as they pinpoint areas of the reconstructed surface, that have

been reconstructed under sub-optimal conditions. Combining them with the mesh-based metrics yield at least another 5-6% increase in the performance of the noise estimation, depending on which mesh-based metrics, they are combined with.

Finally we test the larger context of the proposed metrics for detecting noise on 3D reconstructions, which have significant difference from the data used for capturing the training metrics. This way such robustness can be tested. A wind turbine blade is selected, as their inspection has become of particular research interest. The blade also has a different size, shape and material from all the other tested objects. We demonstrate that we can achieve usable results, without miss-classifying any surface damage as reconstruction noise. This result also shows that the proposed metrics can be used as a form of transfer learning, where a noise detector can be trained on generic widely available data and then used on specialized data, which does not contain a large enough dataset, like wind turbine blade surfaces. The produced results of 0.843 accuracy 0.786 precision and 0.877 recall, show that the same level of quality of noise estimation can be achieved for wind turbine blades, which can be seen as an extended general applicability of the presented research.

The next step in verifying the results of the publication, would be comparing the reconstructed meshes to ground truth of the object, captured with a high resolution scanner. The difference between the two can be used, as a more objective noise ground truth, which can be then used to compare to the estimated noise risk. A look into global deformations in the overall shape of the reconstructed objects, as well as self-occlusions and fractal parts of the objects, can also be used to further introduce more metrics for assessing the risk of noise. Finally, one can also look even more into the influence of the camera specifications on the possibility of noise, such as the use of fixed focus lens versus an automatic focus one, as well as the use of rolling versus a global shutter.

Our future work would build on the results from this paper, by comparing them to both traditional mesh denoising algorithms and newer point cloud and mesh classification methods using convolutional and deep neural networks. For this a larger dataset of SfM object reconstruction is being build, so enough data is present. Finally, it is deemed interesting to look into detecting the illumination levels of the environment and see if they can be used as reliable indicators, as the role of the capturing setup lighting in the presence of noise, requires more research.

## References

- [1] S. Siebert and J. Teizer, "Mobile 3d mapping for surveying earthwork projects using an unmanned aerial vehicle (uav) system," *Automation in*

## References

- Construction*, vol. 41, pp. 1–14, 2014.
- [2] S. Tuttas, A. Braun, A. Borrmann, and U. Stilla, “Acquisition and consecutive registration of photogrammetric point clouds for construction progress monitoring using a 4d bim,” *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 85, no. 1, pp. 3–15, 2017.
- [3] K. Chaiyasarn, T.-K. Kim, F. Viola, R. Cipolla, and K. Soga, “Distortion-free image mosaicing for tunnel inspection based on robust cylindrical surface estimation through structure from motion,” *Journal of Computing in Civil Engineering*, vol. 30, no. 3, p. 04015045, 2015.
- [4] A. Khaloo and D. Lattanzi, “Hierarchical dense structure-from-motion reconstructions for infrastructure condition assessment,” *Journal of Computing in Civil Engineering*, vol. 31, no. 1, p. 04016047, 2016.
- [5] D. Zhang, K. Burnham, L. McDonald, C. Macleod, G. Dobie, R. Summan, and G. Pierce, “Remote inspection of wind turbine blades using uav with photogrammetry payload,” in *56th Annual British Conference of Non-Destructive Testing-NDT 2017*, 2017.
- [6] S. P. Bemis, S. Micklethwaite, D. Turner, M. R. James, S. Akciz, S. T. Thiele, and H. A. Bangash, “Ground-based and uav-based photogrammetry: A multi-scale, high-resolution mapping tool for structural geology and paleoseismology,” *Journal of Structural Geology*, vol. 69, pp. 163–178, 2014.
- [7] Y. Cho and R. Clary, “Application of sfm-mvs photogrammetry in geology virtual field trips,” in *81st annual meeting of Mississippi Academy Sciences, Hattiesburg, MS, USA*, 2017.
- [8] H. Bi, W. Zheng, Z. Ren, J. Zeng, and J. Yu, “Using an unmanned aerial vehicle for topography mapping of the fault zone based on structure from motion photogrammetry,” *International journal of remote sensing*, vol. 38, no. 8-10, pp. 2495–2510, 2017.
- [9] T. P. Kersten and M. Lindstaedt, “Image-based low-cost systems for automatic 3d recording and modelling of archaeological finds and objects,” in *Euro-Mediterranean Conference*. Springer, 2012, pp. 1–10.
- [10] G. Kyriakaki, A. Doulamis, N. Doulamis, M. Ioannides, K. Makantasis, E. Protopapadakis, A. Hadjiprocopis, K. Wenzel, D. Fritsch, M. Klein *et al.*, “4d reconstruction of tangible cultural heritage objects from web-retrieved images,” *International Journal of Heritage in the Digital Era*, vol. 3, no. 2, pp. 431–451, 2014.

## References

- [11] S. W. Hixon, C. P. Lipo, T. L. Hunt, and C. Lee, "Using structure from motion mapping to record and analyze details of the colossal hats (pukao) of monumental statues on rapa nui (easter island)," *Advances in Archaeological Practice*, vol. 6, no. 1, pp. 42–57, 2018.
- [12] K. Thoeni, A. Giacomini, R. Murtagh, and E. Kniest, "A comparison of multi-view 3d reconstruction of a rock wall using several cameras and a laser scanner," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, no. 5, p. 573, 2014.
- [13] J. Schönig and G. Heidemann, "Evaluation of multi-view 3d reconstruction software," in *International conference on computer analysis of images and patterns*. Springer, 2015, pp. 450–461.
- [14] I. Nikolov and C. Madsen, "Benchmarking close-range structure from motion 3d reconstruction software under varying capturing conditions," in *Euro-Mediterranean Conference*. Springer, 2016, pp. 15–26.
- [15] Agisoft, "Metashape (photoscan)," <http://www.agisoft.com/>, 2010, accessed: 2020-06-15.
- [16] Bentley, "Contextcapture," <https://www.bentley.com/en/products/brands/contextcapture>, 2016, accessed: 2020-06-15.
- [17] CapturingReality, "Reality capture," <https://www.capturingreality.com/>, 2016, accessed: 2020-06-15.
- [18] B. Kim and J. Rossignac, "Geofilter: Geometric selection of mesh filter parameters," in *Computer Graphics Forum*, vol. 24, no. 3. Wiley Online Library, 2005, pp. 295–302.
- [19] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa, "Laplacian mesh optimization," in *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*. ACM, 2006, pp. 381–389.
- [20] Z.-X. Su, H. Wang, and J.-J. Cao, "Mesh denoising based on differential coordinates," in *2009 IEEE International Conference on Shape Modeling and Applications*. IEEE, 2009, pp. 1–6.
- [21] X. Lu, W. Chen, and S. Schaefer, "Robust mesh denoising via vertex pre-filtering and l1-median normal filtering," *Computer Aided Geometric Design*, vol. 54, pp. 49–60, 2017.
- [22] P.-S. Wang, Y. Liu, and X. Tong, "Mesh denoising via cascaded normal regression." *ACM Trans. Graph.*, vol. 35, no. 6, pp. 232–1, 2016.

## References

- [23] X. Lu, X. Liu, Z. Deng, and W. Chen, "An efficient approach for feature-preserving mesh denoising," *Optics and Lasers in Engineering*, vol. 90, pp. 186–195, 2017.
- [24] Y. Zheng, H. Fu, O. K.-C. Au, and C.-L. Tai, "Bilateral normal filtering for mesh denoising," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 10, pp. 1521–1530, 2010.
- [25] O. Wasenmüller, G. Bleser, and D. Stricker, "Joint bilateral mesh denoising using color information and local anti-shrinking," 2015.
- [26] G. Lavoué, "A local roughness measure for 3d meshes and its application to visual masking," *ACM Transactions on Applied perception (TAP)*, vol. 5, no. 4, p. 21, 2009.
- [27] K. Wang, F. Torkhani, and A. Montanvert, "A fast roughness-based approach to the assessment of 3d mesh visual quality," *Computers & Graphics*, vol. 36, no. 7, pp. 808–818, 2012.
- [28] R. Song, Y. Liu, R. R. Martin, and P. L. Rosin, "Mesh saliency via spectral processing," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 1, p. 6, 2014.
- [29] J. Guo, V. Vidal, A. Baskurt, and G. Lavoué, "Evaluating the local visibility of geometric artifacts," in *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception*. ACM, 2015, pp. 91–98.
- [30] Y. Ioannou, B. Taati, R. Harrap, and M. Greenspan, "Difference of normals as a multi-scale operator in unorganized point clouds," in *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*. IEEE, 2012, pp. 501–508.
- [31] T. Rabbani, F. Van Den Heuvel, and G. Vosselmann, "Segmentation of point clouds using smoothness constraint," *International archives of photogrammetry, remote sensing and spatial information sciences*, vol. 36, no. 5, pp. 248–253, 2006.
- [32] S. Harwin and A. Lucieer, "Assessing the accuracy of georeferenced point clouds produced via multi-view stereopsis from unmanned aerial vehicle (uav) imagery," *Remote Sensing*, vol. 4, no. 6, pp. 1573–1599, 2012.
- [33] B. Huhle, T. Schairer, P. Jenke, and W. Straßer, "Robust non-local denoising of colored depth data," in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2008, pp. 1–7.
- [34] A. Shiozaki, "Edge extraction using entropy operator," *Computer Vision, Graphics, and Image Processing*, vol. 36, no. 1, pp. 1–9, 1986.

## References

- [35] M. Favalli, A. Fornaciai, I. Isola, S. Tarquini, and L. Nannipieri, "Multiview 3D reconstruction in geosciences," *Computers & Geosciences*, vol. 44, pp. 168–176, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0098300411003128>
- [36] N. D'Amico and T. Yu, "Accuracy analysis of point cloud modeling for evaluating concrete specimens," in *SPIE Smart Structures and Materials+ Nondestructive Evaluation and Health Monitoring*. International Society for Optics and Photonics, 2017, pp. 101 691D–101 691D.
- [37] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer, "A survey of structure from motion\*." *Acta Numerica*, vol. 26, pp. 305–364, 2017.
- [38] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.
- [39] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European conference on computer vision*. Springer, 2006, pp. 430–443.
- [40] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.
- [41] Kodak, *Optical Formulas and Their Applications*. Kodak, 1969.
- [42] M. Marčíš, "Quality of 3d models generated by sfm technology," *Slovak Journal of Civil Engineering*, vol. 21, no. 4, pp. 13–24, 2013.
- [43] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.
- [44] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
- [45] U. Technologies, "Unity," [unity.com](http://unity.com), 2005, accessed: 2020-07-20.
- [46] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise reduction in speech processing*. Springer, 2009, pp. 1–4.
- [47] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

## References

- [48] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.
- [49] D. Zhang, R. Watson, G. Dobie, C. MacLeod, A. Khan, and G. Pierce, "Quantifying impacts on remote photogrammetric inspection using unmanned aerial vehicles," *Engineering Structures*, p. 109940, 2020.
- [50] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PloS one*, vol. 10, no. 3, p. e0118432, 2015.

# Paper H

## High-resolution Structure-from-Motion for Quantitative Measurement of Leading Edge Roughness

Mikkel Schou Nielsen, Ivan Nikolov, Emil Krog Kruse, Jørgen  
Garnæs and Claus Madsen

The paper has been published in the  
*MDPI Energies Journal* Vol. 13(15), 2020.

© 2020 MDPI

*The layout has been revised.*

### **Abstract**

*Over time, erosion of the leading edge of wind turbine blades increases the leading edge roughness (LER). This may reduce the aerodynamic performance of the blade and hence the annual energy production of the wind turbine. As early detection is key for cost-effective maintenance, inspection methods are needed to quantify the LER of the blade. The aim of this proof-of-principle study is to determine whether high-resolution Structure-from-Motion (SfM) has the sufficient resolution and accuracy for quantitative inspection of LER. SfM provides 3D reconstruction of an object geometry using overlapping images of the object acquired with a RGB camera. Using information of the camera positions and orientations, absolute scale of the reconstruction can be achieved. Combined with a UAV platform, SfM has the potential for remote blade inspections with a reduced down-time. The tip of a decommissioned blade with an artificially enhanced erosion was used for the measurements. For validation, replica moulding was used to transfer areas-of-interest to the lab for reference measurements using confocal microscopy. The SfM reconstruction resulted in a spatial resolution of 1 mm as well as a sub-mm accuracy in both the RMS surface roughness and the size of topographic features. In conclusion, high-resolution SfM demonstrated a successful quantitative reconstruction of LER.*

### **H.1 Introduction**

Erosion of wind turbine blades poses a challenge for wind energy operation and maintenance [1]. Erosion of the leading edge (LE) increases the surface roughness and reduces the aerodynamic performance of the blade [1, 2]. As the shape of wind turbine blades is specifically designed to achieve maximum energy efficiency [3], this increased leading edge roughness (LER) may lead to a reduced annual energy production of the wind turbine. Through CFD modelling, several studies have found that even a small degree of LE erosion can lead to 2%-5% loss in annual energy production [4–6]. Severely eroded blades with high levels of LER can experience losses from 8% and up to 25% [4, 7, 8]. As LE erosion over time can develop from small pinholes to large areas of coating delamination [4, 9], early detection of the severity of the erosion is important. At later erosion stages, extensive blade repair may be necessary causing expensive turbine down-time. Thus for early erosion detection, inspection methods for measuring the surface topography of the blade are needed to quantify the LER.

Visual inspection have long been applied for condition-monitoring of wind turbine blades [10]. In recent years, unmanned aerial vehicles (UAV) have received increased interest for remote inspection of wind turbines [11–16] with a lower downtime compared to manual rope-access inspection. From 2D images captured by the UAV, deep learning methods [13, 15] can be used for

detecting damages and erosion on the blades. However, while the 2D information can reveal the presence and location, *quantification* of the blade surface roughness requires high-resolution 3D data.

Structure-from-Motion (SfM) is a camera-based method that provides a 3D reconstruction of an object geometry with a simple, fast and low-cost acquisition [17–19]. Aided by a rapid development of both open source [20, 21] and commercial software solutions [22–24], SfM has found industrial interest in e.g. construction site monitoring [25–27] and infrastructure inspection [28–32]. A lot of research has also been done in the performance of SfM, for different use cases [33, 34]. As input for the SfM reconstruction, overlapping images of the object from different positions and orientations are acquired using a RGB camera. Feature points are extracted and matched between the 2D input images using local feature descriptors such as SIFT [35] or ORB [36]. From the feature points and intrinsic camera parameters, a sparse 3D point cloud as well as the camera positions and orientations are computed. Using information from reprojected camera views, further points can be added to create a dense point cloud, which can be further meshed [37, 38].

The accuracy of a SfM reconstruction is influenced by a number of factors. Since SfM depends on triangulation of feature points, the accuracy is affected by the angular coverage of the acquired images [19, 39] and scales with the capturing distance from camera to object [38, 40, 41]. Furthermore, a sufficient texture level is required for enough distinct features on the object surface to be tracked from image to image [19, 42]. Low texture regions may result in empty regions of the point cloud [43]. To evaluate the accuracy, the SfM reconstruction is typically compared to another optical technique such as a LiDAR or laser scanner. This can be done either by direct point-to-point comparison with the SfM point cloud [19, 44, 45] or raster-to-raster comparison of digital elevation models (DEM) [46, 47]. Either way, the comparison is influenced by the measurement uncertainty of the reference points [47]. Common metrics for reporting the accuracy are the standard deviation (SD) [17, 33, 44] and root mean square deviation (RMSD) [41, 43, 46].

Within wind energy, SfM has previously been investigated for 3D reconstruction of blade geometries [11, 16]. However, these studies did not have a sufficient resolution to reconstruct the surface topography directly and rather used the color texture to identify damages. With high-resolution SfM, a point-sampling distance below 0.1 mm/pixel can be achieved which allows for reconstruction of the surface roughness [48, 49].

In this proof-of-principle study, we investigate the potential of high-resolution SfM in quantitative inspection of wind turbine blades. We envisage a scenario where an UAV carrying a high-end RGB camera is capturing images of the LE of blades. Using these images, a SfM reconstruction of (parts of) the LE is performed from which quantitative measures of the LER can be extracted. The study seeks to answer two main questions. Firstly, to demonstrate whether

a sufficient resolution can be achieved to reconstruct the LER of a blade. Secondly, what is the performance of high-resolution SfM in providing quantitative measures of the surface topography of the LER. A mock-up of an eroded blade was fabricated by artificially enhancing the LER of the tip of a decommissioned blade. The SfM capturing was done using a handheld camera and in an outdoor environment to mimic realistic inspection conditions. In the high-resolution acquisition, the images were acquired from a distance of roughly 2 meters using a 300 mm lens. We believe these conditions to be representative of what the envisaged UAV inspection scenario might operate with.

For evaluating the accuracy of the SfM reconstruction, selected areas on the blade surface were extracted from the point cloud and converted to a DEM. Replicas of the same areas on the blade surface were made using replication moulding and transferred to the lab. Replication moulding is a demonstrated method for transferring hard-to-access surface topographies to a substrate suitable for microscopy measurements [50]. In the replication of surface roughness, accuracies at the sub-micrometer level have been demonstrated using elastomer replica materials [51–56]. Using confocal microscopy (CM) measurements of the replicas, a DEM was created for direct raster-to-raster comparison to the SfM reconstruction. The resolution of the SfM reconstruction was evaluated using Fourier analysis and RMSD calculation. For validation of the resolution analysis, a model was constructed by reducing the resolution of the reference DEM and adding noise. This model DEM was then compared to the SfM DEM. Finally, the quantitative performance in measuring LER was evaluated using surface roughness parameters and topographic feature sizes.

## H.2 Methods and Materials

### H.2.1 Blade Mock-up

A decommissioned wind turbine blade was available for the experimental setup. The blade had been used in a modern 2 MW pitch-regulated wind turbine. Span-wise, the outer two meters of the blade that already had some erosion was used. To better resemble the examples of severe LE erosion experienced from field inspections [4, 8, 9], the erosion was artificially increased by sandblasting the LE. At this level of erosion, large areas of laminate are exposed along the LE with depths of 1-3 mm. Severe erosion was chosen for this study for two main reasons. Firstly, depths of these magnitude are at the order where the aerodynamic performance is significantly impacted. A study by [57] found that the critical height of roughness for lowering the maximum lift of wind turbine blades was above 1 mm. Secondly, a large surface rough-

ness represents a good pass/fail test of the feasibility of high-resolution SfM. If the resolution was not sufficient for resolving large erosion structures, it would not function for less eroded surfaces either.

For inspection of a wind turbine in operation, we envision that the turbine is stopped with the inspected blade in a vertical position. To simulate this, the blade was mounted vertically in a gantry, which was welded together for the purpose of this work. To include the effect of oscillations, chain links were used to fix the blade mock-up to the gantry, which let the blade segment move freely in the wind. The height of the gantry was 5 meters, resulting in a distance of 5 meters from the very tip of the blade to the ground. The blade setup on the gantry, together with the scissor lift used to capture all the data for this paper can be seen in Figure H.1.



**Fig. H.1:** The wind turbine blade segment positioned on the built gantry, together with the scissor lift used for capturing image and replica data.

## H.2.2 SfM Capturing Conditions

The image capturing process of the proof-of-principle study was done in an outdoor environment to ensure realistic capturing conditions. A commercial DSLR camera (Canon 5Ds) with a variable zoom lens (Canon 70-300 f/4-5.6L IS USM) was used with the focal length fixed at 300 mm. Camera parameters and settings are summarized in Table H.1. As the capturing was done outdoors, a number of prerequisites need to be taken into account:

- The natural illumination can change between images.

## H.2. Methods and Materials

- The sides of the blade mock-up may not be evenly illuminated.
- Wind can cause oscillations of the blade mock-up, which can change its position and orientation compared to previous images.

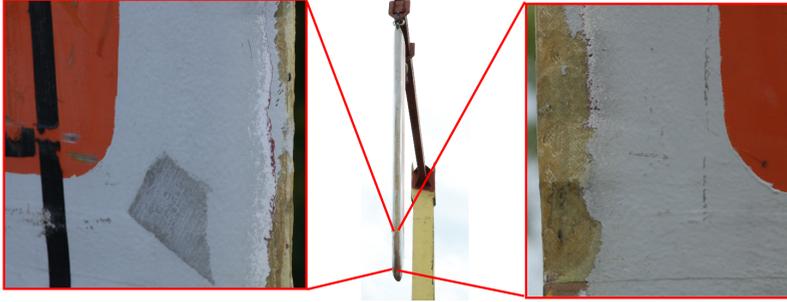
Since the accuracy of the SfM reconstruction depends on the stability of light conditions, camera settings should be robust to environmental changes in light direction and intensity. In addition, the settings should take into account the possible motion of the blade. The chosen ISO, shutter, and aperture settings are shown in Table H.1. They represent what we believe to be reasonable compromises between exposure for outdoor conditions, becoming less sensitive to motion blur (shutter) and not having to worry too much about too shallow depth-of-field (aperture).

**Table H.1:** Camera parameters and settings for the outdoor capturing setup.

<b>Camera Parameters</b>	<b>Values</b>
ISO	800
Shutter Speed [sec]	1/200
Aperture	f/16
Focal length [mm]	300
Image size [pixels]	8688x5792
Sensor pixel size [ $\mu\text{m}$ ]	4.14
<b>Capturing setup</b>	
Distance to blade [m]	2
Angular spacing [ $^\circ$ ]	10
Capturing bands	3
No of images	57
GSD [ $\mu\text{m}/\text{pixel}$ ]	27

For the initial proof-of-principle study, a manual and hand-held image capturing was performed. A part of the wind turbine blade was chosen that contained a variation in surface topography across the leading edge - from very rough damaged areas to smoother clean areas. The part of the blade chosen for 3D reconstruction is shown in Figure H.2.

A semi-circular 180-degree capturing pattern is used for the image capturing. This capturing method was shown by [34] and [58] to produce high accuracy reconstructions, while also minimizing the number of required images. Three horizontal semi-circular bands each with 19 images were acquired giving a total of 57 images. To ensure enough vertical separation between the horizontal bands, the first band was taken from the ground level looking toward the suspended blade. To capture the other two bands a moving scissor lift was used. The captured positions can be seen in Figure H.3b. This way the blade surface could be captured from different positions and angles both in



**Fig. H.2:** Testing blade together with the region chosen for reconstruction. The zoomed-in parts are of the two sides of the blade

horizontal and vertical direction, ensuring maximum cover. The semi-circles were centered around the wind turbine blade with a distance of 2 meters from camera to blade. The capturing settings are summarized in Table H.1.

For a fixed focal length, the capturing distance determines the ground sampling distance (GSD), i.e. the spatial size on the object that each pixel in a captured image covers. Using a pinhole camera model, the GSD can be calculated as the camera sensor pixel size  $p$  multiplied by the ratio of the distance between camera and surface  $D$  over the focal length  $f$  as shown in equation H.1.

$$GSD = \frac{D}{f}p \quad (\text{H.1})$$

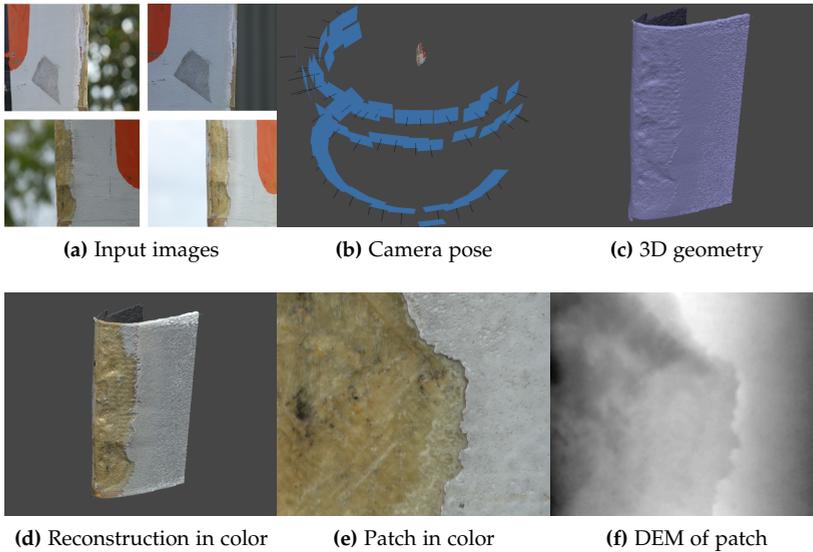
With the used settings in the study, the GSD was  $27 \mu\text{m}/\text{pixel}$  which corresponds to approximately 36 pixels/mm on the blade surface.

### H.2.3 SfM Reconstruction

For SfM reconstruction, the commercial stand-alone software package Agisoft Metashape by [22] was used. It was selected as it has previously demonstrated a high accuracy compared to other state of the art solutions, while being robust against sub-optimal capturing conditions [34]. The pipeline from input images, 3D reconstruction and extraction of depth map patches is visualized in Figure H.3. An overview of the process is given below.

The captured images H.3a were imported to Metashape and a triangulation, feature extraction and matching step were performed to find the camera positions and key feature points from the input images H.3b. From these positions and feature points, a sparse point cloud was formed. Next in the

## H.2. Methods and Materials



**Fig. H.3:** Pipeline for 3D reconstruction using SfM. H.3a) Initially, images were acquired at every 10 degrees of a half circle around the wind turbine blade at three different heights and tilt angles. H.3b The camera pose of the images and points on the object surface were then calculated. The reconstructed surface geometry without H.3c and with color H.3d. Extracted patch from the reconstruction shown as H.3e (color) texture and H.3f resulting DEM.

reconstruction process a dense point cloud was created and meshed into a triangle mesh H.3c. Finally a (color) texture is build from the visual data from the input images H.3d.

To establish an absolute scale in the SfM reconstruction, the known camera positions and distance from camera to blade surface were utilized. The method presented in [59] was followed. The scale was calculated using a least squares transformation estimation between the reconstructed camera positions and the manually measured positions in the real world.

For evaluating the SfM reconstruction, three areas R1, R2 and R3 were selected for comparison to reference microscopy measurements. The areas were chosen to include distinctive surface topography features and cover the boundary between intact coating and damaged surface. For each area, a digital elevation model (DEM) was created from the reconstruction using the following pipeline. First, for further processing and analysis of the mesh the reconstruction was imported to the software CloudCompare [60]. For each area, a patch of roughly 35 mm  $\times$  35 mm was created from the main reconstructed point cloud. The patches were oriented with the Z axis perpendicular to the mesh surface, and were rasterized into a DEM of the surface topography H.3f. This was done by an interpolation of the point-cloud

points to a map with equidistant point spacing and using the average  $z$ -values of each grid space. The resulting pixel size was chosen to be  $13,3\mu\text{m}$  to match the reference microscopy images.

## H.2.4 Replica Moulding

Replication was performed for each of the R1, R2 and R3 areas described in section H.2.3. As a replication material with a fast curing time and resolution down to  $0.1\mu\text{m}$ , Repliset T3 by Struers [61] was selected. In previous studies, the replication of surface textures using Repliset has achieved a sub-micrometer accuracy [55, 62]. The RepliSet T3 is a black two-part silicone rubber which consists of a polymer and curing agent. For replication, the two parts were pushed out of the cartridge, mixed in a static-mixing nozzle and applied onto the blade surface H.4a. Immediately after application, backing paper was placed on top of the mixture and attached by applying a small force as shown in Figure H.4b. The mixture set for 15 minutes and then the replica was removed from the blade surface by hand.



**Fig. H.4:** Replication of an area on the blade mock-up. Figure H.4a illustrates the application of the combined polymer and curing agent from a static-mixing nozzle. Figure H.4b shows the backing paper being attached by applying a gentle force to the replication material.

## H.2.5 Confocal microscopy

Confocal microscopy (CM) was used to produce reference DEMs of the R1, R2 and R3 areas. The three replica of the blade surface were measured using a calibrated PLU NEOX confocal microscope by Sensofar [63]. For each replica, an extended area of approx.  $35\text{ mm} \times 35\text{ mm}$  was measured by stitching around 400 individual images. A  $\times 5$  magnification objective with an NA of 0.15 was used. For each image, a  $4 \times 4$  binning was used resulting in a final pixel size of  $13.3\mu\text{m}$ . To ensure a superior resolution for the CM measurement, the pixel size was kept smaller than the GSD of the SfM reconstruction. The vertical step size ( $z$ -axis) used was  $12\mu\text{m}$ . The 3D surface

reconstruction, stitching and creation of a DEM were performed using the proprietary SensoSCAN software.

The sensitivity of the CM microscope in the vertical direction ( $z$ -axis) was calibrated using a set of step height transfer standards. Traceability was ensured through calibration of the standards by e.g. an AFM equipped with laser interferometer. The amplification coefficient of the  $z$ -axis had a relative uncertainty lower than 3%.

## H.2.6 Image processing and data analysis

The main software programs used for the surface topography analysis were the Scanning Probe Image Processor (SPIP) [64] version 6.6.3 as well as custom scripts in MATLAB version 2019b. SPIP is an image processing program with special tools for accurate characterization of image structures.

Initially using SPIP, each SfM and reference DEM were levelled by subtracting a least-squares parabola fit from the overall shape. This way the long wavelength curvature of the surface was removed, while the short wavelength surface roughness could be preserved. Then for each area, the SfM DEMs were co-registered using a Fourier correlation approach in MATLAB.

From the co-registered DEMs, geometrical quantities were extracted from both SfM and microscopy reference. The chosen quantities are the depth and height of topographic features.

### SfM reconstruction quality

To evaluate the quality of the SfM reconstruction, two parameters were chosen; The instrument transfer function at 50% value (ITF50) and the maximum value of the cross-correlation function ( $CCF_{\max}$ ).

The ITF50 value is a measure of the spatial sharpness, which is analogous to the MTF50 value of the modulation transfer function. ITF50 is found as the spatial wavelength at which the instrument response is half the value of the reference. The definition of ITF is shown in (H.2) [65]. For the calculation, a region on the surface containing a height is selected. For each line across the height step, the ratio of the 1D Fourier transforms of the instrument function and reference is calculated. The ITF is found as the mean of all lines in the region.

$$ITF(f) = \left\langle \frac{\left| \int_{-\infty}^{\infty} z(x, y) e^{-i2\pi f x} dx \right|}{\left| \int_{-\infty}^{\infty} z_{ref}(x, y) e^{-i2\pi f x} dx \right|} \right\rangle_y \quad (H.2)$$

$CCF_{\max}$  has a value between 0 and 1 and describes the spatial similarity of a set of co-registered measurement and reference topographies. If the

measurement is very close to the reference, the value of  $CCF_{\max}$  will be close to 1.  $CCF_{\max}$  is found as the maximum of the normalized 2D cross-correlation function [66].

$$f_{CCF}(t_x, t_y) = \frac{\iint_A z(x, y) z_{ref}(x - t_x, y - t_y) dx dy}{\sqrt{\iint_A z^2(x, y) dx dy \iint_A z_{ref}^2(x, y) dx dy}} \quad (H.3)$$

### Surface Roughness

Prior to performing the surface roughness analysis, an S-filter of  $25 \mu m$  and an L-filter of  $10 mm$  were applied to the DEM in accordance with [67]. A plane was chosen as reference surface using a least-squares linear fit to the height values of the DEM. The following areal surface roughness parameters as described in [68] were calculated:  $S_q$ ,  $S_{dq}$  and  $S_{al}$ . These parameters were chosen as they describe different and complementary features of the surface topography as explained below. The analysis was performed using the "Roughness Analysis" tool of the SPIP application software.

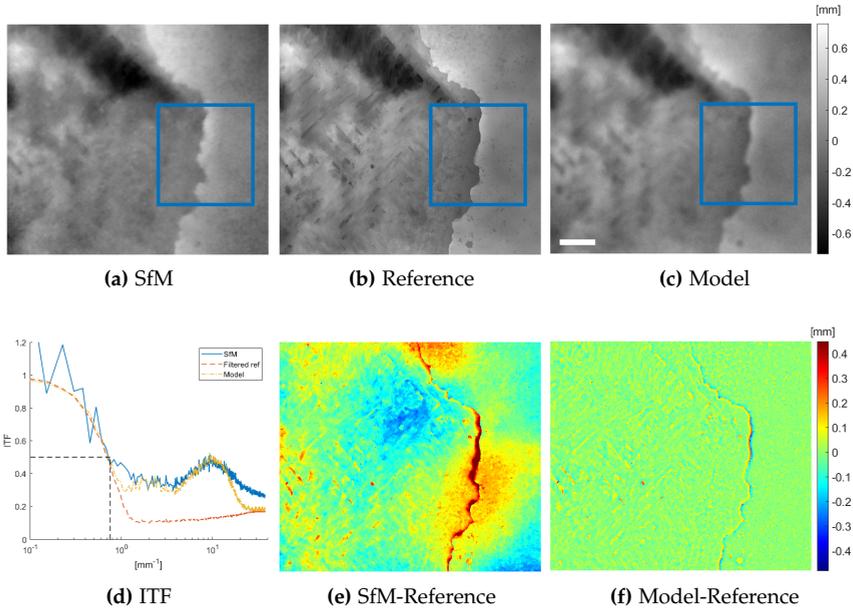
- $S_q$  is the root mean square height of the  $z$ -values with respect to the reference surface.  $S_q$  describes the overall height variation of the DEM.
- $S_{dq}$  is the root mean square of the surface gradient. As  $S_{dq}$  depends on variations in the local slope, it is sensitive to the short wavelength components of the surface topography.
- $S_{al}$  is the auto-correlation length, and is a measure of the spatial distance at which the surface texture becomes statistically different.  $S_{al}$  is calculated as the minimum distance in frequency space at which the auto-correlation function decays to 0.2 in value. Therefore,  $S_{al}$  contains information on the long wavelength components of the surface topography.

## H.3 Results

The performance of the SfM reconstruction is illustrated in Figure H.5. In H.5a and H.5b, the SfM and reference DEM of replication area R1 are shown. Both have been processed as described in section H.2.6. The blue box indicates the subregion used for the ITF analysis. As seen in panel H.5a, the SfM DEM captures the main topographic features although the resolution is less than for the reference in H.5b. While short wavelength topography variations are missing, holes, edges and the larger glass-fiber structures are visible in the SfM DEM.

The ITF function was calculated from the SfM and Reference DEM of area R1 as described in section H.2.6. The ITF was not calculated for R2 and R3 as

### H.3. Results



**Fig. H.5:** H.5a Reference, H.5b SfM and H.5c model DEM of replication area R1. The blue box indicates the area used for the calculation of ITF50. The scalebar is 5 mm. H.5d ITF for SfM and two model curves based on filtering the reference DEM with and without noise added. H.5e-H.5f Residual of SfM and model DEM with respect to reference DEM.

no height step was present in these areas. In H.5d, the ITF function for SfM is shown (blue solid line) with the 50% value indicated in dashed black lines. As stated in Table H.2, the ITF50 spatial wavelength was 1.3 mm.

In order to validate the shape of the SfM ITF, a model was developed based on the reference DEM. First, the reduced resolution of the SfM reconstruction was approximated by applying a Gaussian low-pass filter to the reference DEM. A filter with a FWHM of 0.65 mm was used to give the model DEM the same ITF50 value as the SfM. As seen in Figure H.5d, the ITF of the filtered reference (dashed red line) matches the long wavelength values of the SfM ITF. However, at higher frequencies the filtered reference has lower values than the SfM. In a second step, structured noise centered at 0.5 mm and 0.1 mm wavelengths was added (dash-dotted yellow line). The two noise components was constructed through Gaussian low-pass filtering of Gaussian noise with an amplitude selected to match the SfM ITF. In H.5c, the model DEM using Gaussian filter and noise added is shown for area R1. Similarly, a model DEM was created for both area R2 and R3 using the same Gaussian FWHM and noise settings.

In Table H.2,  $\text{CCF}_{\max}$  and RMSD values for both SfM and model DEM

are shown for all replication areas. The  $CCF_{\max}$  values for the model were close to 1 for all areas. For SfM, the values were  $>0.9$  for both R2 and R3 indicating a very close horizontal spatial resemblance. A slightly smaller value was found for R1. Overall, the  $CCF_{\max}$  values indicate that an accurate scaling of SfM was obtained. As seen in the table, the RMSD values for SfM are between 0.1-0.2 mm, which is several times larger than the model values between 0.03-0.04 mm. This discrepancy is illustrated in the residuals shown in Figure H.5e and H.5f. While large differences are observed near sharp edges for both model and SfM DEM, the SfM residuals also contain a waviness that accounts for the larger RMSD value. The waviness has a wavelength in the order of 10 mm which shows up as large variations in the long wavelength part of the ITF in Figure H.5d.

**Table H.2:** Quantitative values for evaluating the SfM reconstruction.  $CCF_{\max}$  and RMSD are included for all three areas R1 to R3 while ITF50 was only calculated for R1.

Replication area	R1	R2	R3
ITF50 [mm]	1.3		
$CCF_{\max}$ , SfM	0.86	0.95	0.93
$CCF_{\max}$ , Model	0.98	0.99	0.995
RMSD, SfM [mm]	0.099	0.12	0.21
RMSD, Model [mm]	0.038	0.036	0.031

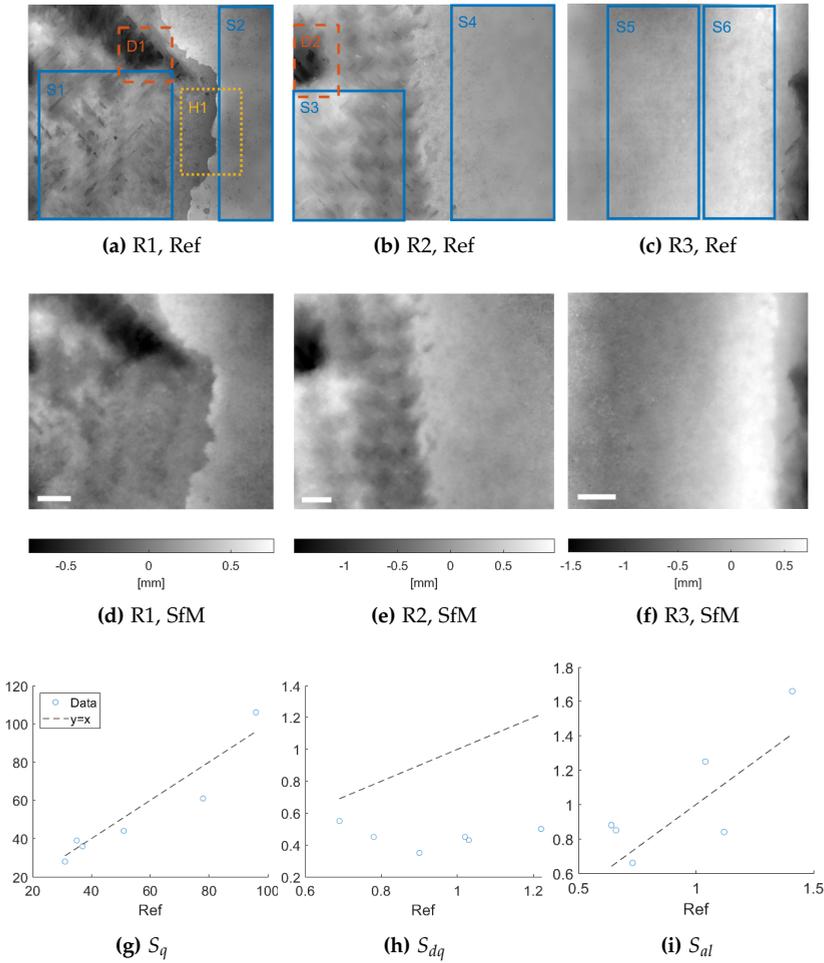
To evaluate the surface roughness of the SfM reconstruction, two regions within each of the R1, R2 and R3 areas were selected as shown with blue solid lines in Figure H.6a to H.6c. Of the six regions named S1 to S6, an extensive erosion of S1 and S3 resulted in a topography dominated by glass-fiber structures while S2, S4, S5 and S6 still had an intact surface coating.

The roughness parameters  $S_q$ ,  $S_{dq}$  and  $S_{al}$  were calculated for each of the S1 to S6 regions as described in section H.2.6.

The roughness values for both SfM and reference are shown in Table H.3 and illustrated in scatter plots in Figure H.6g to H.6i. Overall, both SfM and reference values show a larger roughness for the eroded S1 and S3 regions than the regions with intact coating. However, when comparing the three roughness parameters on the scatter plots, some differences are clear. While the SfM values for  $S_q$  and  $S_{al}$  vary within around 10%-20% of the reference values. the SfM values for  $S_{dq}$  are systematically lower than the reference by around 50%. The absolute RMSD deviations for S1 to S6 were 9  $\mu\text{m}$  for  $S_q$ , 0.5 for  $S_{dq}$  and 0.2 mm for  $S_{al}$ .

Three distinctive topographic features were selected in the R1 and R2 areas; Two depressions D1 and D2 (red dashed lines) and a height step H1 (yellow dotted lines) as shown in Figure H.6a and H.6b. For all features, the depth and height measurements for the SfM DEM are close to the reference

### H.3. Results



**Fig. H.6:** H.6a-H.6c Reference and H.6d-H.6f SfM DEMs of area R1, R2 and R3, respectively, with prior image processing as described in section H.2.6. Surface roughness regions S1 to S6 are indicated with solid blue lines. Depressions D1, D2 and ridge H1 are indicated with dashed red and dotted yellow lines, respectively. The scalebar is 5 mm. H.6g-H.6i Scatter plots of SfM and reference values for  $S_q$ ,  $S_{dq}$  and  $S_{al}$ , respectively. The dotted line indicates where SfM values are equal to reference values.

as shown in Table H.3. The relative deviations between SfM and reference are less than 16%, and the absolute deviations were less than 0.2 mm with an RMSD of 0.1 mm.

**Table H.3:** Quantitative values for surface roughness and topography features. The roughness parameters  $S_q$ ,  $S_{dq}$  and  $S_{al}$  for regions S1 to S6 were calculated using an S-filter of  $25\ \mu\text{m}$  and an L-filter of  $10\ \text{mm}$ . The depth for depression areas D1 and D2 and height for ridge area H1.

Replication area	R1		R2		R3	
	S1	S2	S3	S4	S5	S6
$S_q$ , Ref [ $\mu\text{m}$ ]	78	31	96	35	37	51
$S_q$ , SfM [ $\mu\text{m}$ ]	61	28	106	39	36	44
$S_{dq}$ , Ref	1.03	0.78	1.02	1.22	0.69	0.90
$S_{dq}$ , SfM	0.43	0.45	0.45	0.50	0.55	0.35
$S_{al}$ , Ref [ $\text{mm}$ ]	1.04	1.12	1.41	0.66	0.73	0.64
$S_{al}$ , SfM [ $\text{mm}$ ]	1.25	0.84	1.66	0.85	0.66	0.88
	D1	H1	D2			
$\Delta z$ , Ref [ $\text{mm}$ ]	0.57	0.34	1.51			
$\Delta z$ , SfM [ $\text{mm}$ ]	0.66	0.30	1.68			

## H.4 Discussion

The reconstructed SfM displayed a high sharpness and resolution. From the ITF50 value, we have that features down to 1.3 mm appear sharp. Conversely, the Gaussian FWHM of 0.65 mm from the model DEM gives a measure of the spatial resolution, i.e. the smallest distinguishable features. The resolution of around 1 mm is one to two orders of magnitude lower than the GSD of  $27\ \mu\text{m}$ , which is in line with previous high-resolution SfM studies [48, 49].

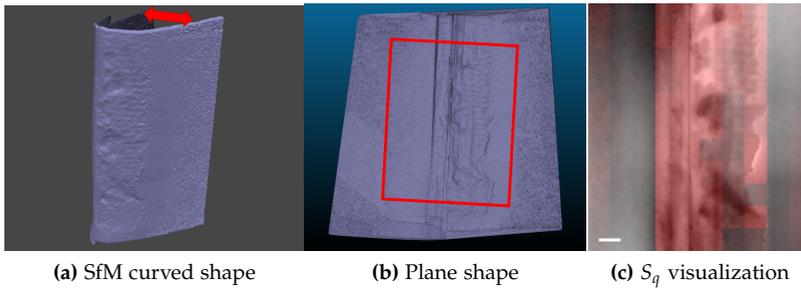
The high value of  $\text{CCF}_{\text{max}}$  for the R2 and R3 areas shows a good spatial resemblance between SfM and reference measurements. The slightly lower value for R1 could either indicate an insufficient resolution or an imperfect co-registration. Since the  $\text{CCF}_{\text{max}}$  values for the model DEM were close to 1, the resolution seems sufficient to preserve the topographic features. The accuracy in co-registering the DEMs could be limited by the replication moulding. While the replica ensures a high replication accuracy of the surface roughness, the overall shape is not preserved when demoulding the replica. Although a levelling was applied, a waviness was still observed in the residual of the SfM DEM with respect to the reference as seen in Figure H.5e. Nonetheless, as the waviness had a wavelength of 10 mm it did not impact the ITF50 value of 1.3 mm.

Furthermore, as indicated by the model DEM, a rather high noise level was present in the SfM DEM. Some of this may originate from the point cloud densification or the interpolation when creating the DEM. Varying light intensity may also affect the reconstruction as reported by [11]. Further studies are needed to determine the potential for reducing the noise level.

The surface roughness analysis show relatively good results for SfM mea-

surement of both  $S_q$  and  $S_{al}$  parameters. The resolution of the SfM reconstruction was sufficient as both  $S_q$  and  $S_{al}$  are most sensitive to the low spatial frequencies, i.e. structures larger than 1 mm. Similarly, the topographic features D1, H1 and D2 had large spatial widths which ensured good results for the measured depths and heights. In contrast, a poor result was seen for the  $S_{dq}$  parameter which is sensitive to high spatial frequencies, i.e. structures smaller than 1 mm.

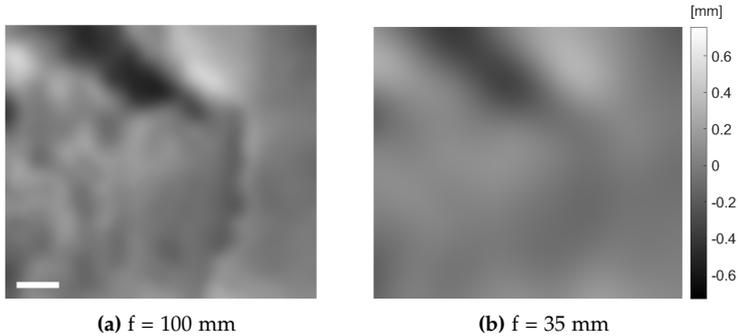
The potential for using the  $S_q$  parameter in quantitative characterization of LER is illustrated in Figure H.7. First, the SfM reconstruction was unfolded to a flat shape and extracted as a DEM using CloudCompare. A region centered on the LE was selected as indicated with a box in Figure H.7b. For each 10 mm  $\times$  10 mm square in the region, the  $S_q$  parameter was calculated and visualized in red in Figure H.7c. The strength of the red color indicates the  $S_q$  value in each square with a lower bound of 20  $\mu\text{m}$  (no color) and upper bound of 100  $\mu\text{m}$  (color saturated). As seen, the  $S_q$  values are low for areas with the coating still intact, and high along the eroded leading edge. This indicates the potential for high resolution SfM for quantitative inspection of LER.



**Fig. H.7:** Visualization of LER. The curved blade geometry seen in H.7a was unfolded to a flat shape shown in H.7b from which a region centered on the LE was selected as indicated by the box. For each 10 mm  $\times$  10 mm square in the region, the  $S_q$  parameter was calculated. In H.7c, the strength of the red color indicates the  $S_q$  value in each square. The scalebar is 10 mm.

For quantitative inspection of the blade erosion, the resolution of the SfM reconstruction needs to match the size of erosion structures, i.e. pits, gauges and delamination. From inspection reports of LE erosion structures, [4] considered pits and gouges with widths down to 0.5 mm and depths from 0.5 mm to 3.8 mm. In the study by [9], widths down to 2 mm and depths from 0.1 mm to 1 mm were investigated. In both studies, the delamination covered tens of millimeters in width and 1-3 mm in depth. The lower end of these feature sizes correspond very closely to the obtained resolution of  $\tilde{1}$  mm. Conversely, as the results of the SfM measurements of topographic features D1, H1 and D2 showed, depths from 0.3 mm - 1.5 mm could be successfully

measured using high-resolution SfM.



**Fig. H.8:** Model DEM of a SfM acquisition of area R1 with H.8a 100 mm and H.8b 35 mm focal length. The model DEMs were created by Gaussian low-pass filtering of the R1 reference DEM of Figure H.5b with FWHM of 1.9 mm and 5.5 mm, respectively. The scalebar is 5 mm.

Had the SfM resolution been lower by e.g. using a shorter focal length, the surface roughness and smaller topographic features would not have been visible. This is illustrated in Figure H.8 by a model DEM of a SfM acquisition of area R1 with a 2 m capturing distance using a 100 mm (H.8a) or 35 mm (H.8b) focal length. These settings correspond to a GSD of 0.08 mm and 0.24 mm, respectively. The model DEMs were created from the R1 reference DEM of Figure H.5b by applying a Gaussian low-pass filter with a FWHM of 1.9 mm and 5.5 mm, respectively. As seen, already for the 100 mm focal length, the glass fiber structures are becoming blurred. For the model of a 35 mm focal length, even the topographic features appear blurred.

In previous studies which applied SfM to reconstruct blade surface, the low resolution would have made a quantification of LER infeasible. In comparison, the settings used by [11, 16] resulted in a GSD of around 0.3 mm which corresponds to the model in Figure H.8b. Rather than quantifying the surface topography, they relied on the texture of the reconstruction to locate damages on the blade surface. An advantage of using a lower resolution is that a larger surface area of the turbine blade can be covered in a single reconstruction. Applying high-resolution SfM to reconstruct the full length of a blade would require a very long inspection time and result in a challenging amount of data.

For full blade inspection, 2D images with even lower resolution can be applied which require fewer image acquisitions and a lower acquisition time. However, in this approach the absolute geometry is not obtained, and the LER is not quantified. Instead other methods would be needed to indicate the presence and location of LER such as the deep learning approach used by [13]. In many ways, the proposed high-resolution SfM is complementary to

this deep learning approach. By combining both, an initial inspection using low-resolution 2D images would indicate the location of LER on the blades. Afterwards, high-resolution SfM could be applied to quantify the severity of the located erosion, which could be used to estimate the aerodynamic impact. Furthermore, if these inspections were combined with a probabilistic model such as a dynamic Bayesian network model [69], the development of the erosion in time could be estimated. This would provide an input for when to conduct repairs on the blade.

## H.5 Conclusion

This proof-of-principle study demonstrated the successful application of high-resolution SfM to quantify the surface roughness of a decommissioned turbine blade. To better resemble the LE erosion observed from inspections, a severe level of erosion with a large area of delamination was applied to the blade. To mimic realistic inspection conditions, the blade was hanged vertically in an outdoor setting, and the SfM image acquisition was conducted hand-held to ensure a level of vibrations. Using a 300 mm focal length and 2 m distance from the blade, a 1 mm spatial resolution of the SfM reconstruction was obtained.

To validate the SfM scan, smaller regions of interest were transferred to the lab using replication moulding and measured with confocal microscopy. From the co-registered SfM and reference regions, a number of surface roughness parameters and topographic feature size were extracted. The quantitative results of surface roughness and topographic feature sizes displayed sub-mm accuracies. Compared to the reference, the RMSD value was  $9 \mu\text{m}$  for the  $S_q$  roughness using an S-filter of 0.025 mm and L-filter of 10 mm, while the RMSD value was 0.1 mm for the depths and heights of topographic features. The results demonstrate the potential for using high-resolution SfM for quantitative measurement of LER on wind turbine blades. Quantitative measurements of LER from blades in operation could aid in creating more realistic CFD models and improve blade inspections.

In future work, a high-resolution SfM inspection using a UAV should be carried out on the blade of a wind turbine in operation. The camera would be mounted in a gantry on the UAV platform to allow for the same poses relative to the blade as in the current study. These settings would allow a more thorough investigation of the effects of vibrations from UAV platform and turbine on the image acquisition and the quality of the 3D reconstructions. In addition, the sensitivity of high-resolution SfM towards surface roughness should be investigated further through measurements on blade surfaces of varying erosion severity. Further studies are also needed on the influence of the texture and color contrast of the blade surface on the quality of the

reconstructed surface details.

## References

- [1] I. F. Zidane, K. M. Saqr, G. Swadener, X. Ma, and M. F. Shehadeh, "On the role of surface roughness in the aerodynamic performance and energy conversion of horizontal wind turbine blades: a review," *International Journal of Energy Research*, vol. 40, no. 15, pp. 2054–2077, dec 2016. [Online]. Available: <http://doi.wiley.com/10.1002/er.3580>
- [2] E. White, D. Kutz, J. Freels, J. Monschke, R. Grife, Y. Sun, and D. Chao, "Leading-Edge Roughness Effects on 63(3)-418 Airfoil Performance," in *49th AIAA Aerospace Sciences*. American Institute of Aeronautics and Astronautics (AIAA), jan 2011.
- [3] K. Yang, "Geometry design optimization of a wind turbine blade considering effects on aerodynamic performance by linearization," *Energies*, vol. 13, no. 9, p. 2320, 2020.
- [4] A. Sareen, C. A. Sapre, and M. S. Selig, "Effects of leading edge erosion on wind turbine blade performance," *Wind Energy*, vol. 17, no. 10, pp. 1531–1542, oct 2014. [Online]. Available: <http://doi.wiley.com/10.1002/we.1649>
- [5] C. M. Langel, R. Chow, O. F. Hurley, C. C. P. Van Dam, D. C. Maniaci, R. S. Ehrmann, and E. B. White, "Analysis of the Impact of Leading Edge Surface Degradation on Wind Turbine Performance," in *33rd Wind Energy Symposium*. Reston, Virginia: American Institute of Aeronautics and Astronautics, jan 2015. [Online]. Available: <http://arc.aiaa.org/doi/10.2514/6.2015-0489>
- [6] W. Han, J. Kim, and B. Kim, "Effects of contamination and erosion at the leading edge of blade tip airfoils on the annual energy production of wind turbines," *Renewable Energy*, vol. 115, pp. 817–823, jan 2018.
- [7] M. Darbandi, A. Mohajer, A. Behrouzifar, R. Jalali, and G. E. Schneider, "Evaluating the effect of blade surface roughness in megawatt wind turbine performance using analytical and numerical approaches," in *10th International Conference on Heat Transfer, Fluid Mechanics and Thermodynamics*, no. July, 2014, pp. 800–805. [Online]. Available: <https://repository.up.ac.za/handle/2263/44584>
- [8] M. Schramm, H. Rahimi, B. Stoevesandt, and K. Tangager, "The Influence of Eroded Blades on Wind Turbine Performance Using Numerical Simulations," *Energies*, vol. 10, no. 9, p. 1420, sep 2017. [Online]. Available: <http://www.mdpi.com/1996-1073/10/9/1420>

## References

- [9] N. Gaudern, "A practical study of the aerodynamic impact of wind turbine blade leading edge erosion," *Journal of Physics: Conference Series*, vol. 524, p. 012031, jun 2014.
- [10] P. Tchakoua, R. Wamkeue, M. Ouhrouche, F. Slaoui-Hasnaoui, T. Tameghe, and G. Ekemb, "Wind Turbine Condition Monitoring: State-of-the-Art Review, New Trends, and Future Challenges," *Energies*, vol. 7, no. 4, pp. 2595–2630, apr 2014. [Online]. Available: <http://www.mdpi.com/1996-1073/7/4/2595>
- [11] L. Wang and Z. Zhang, "Automatic detection of wind turbine blade surface cracks based on uav-taken images," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 9, pp. 7293–7303, 2017.
- [12] L. Peng and J. Liu, "Detection and analysis of large-scale wt blade surface cracks based on uav-taken images," *IET Image Processing*, vol. 12, no. 11, pp. 2059–2064, 2018.
- [13] A. Shihavuddin, X. Chen, V. Fedorov, A. Nymark Christensen, N. Andre Brogaard Riis, K. Branner, A. BJORHOLM DAHL, and R. Reinhold Paulsen, "Wind Turbine Surface Damage Detection by Deep Learning Aided Drone Inspection Analysis," *Energies*, vol. 12, no. 4, p. 676, feb 2019. [Online]. Available: <http://www.mdpi.com/1996-1073/12/4/676>
- [14] P. Durdevic, D. Ortiz-Arroyo, and Z. Yang, "Lidar assisted camera inspection of wind turbines: Experimental study," in *2019 1st International Conference on Electrical, Control and Instrumentation Engineering (ICECIE)*. IEEE, 2019, pp. 1–7.
- [15] Y. Wang, R. Yoshihashi, R. Kawakami, S. You, T. Harano, M. Ito, K. Komagome, M. Iida, and T. Naemura, "Unsupervised anomaly detection with compact deep features for wind turbine blade images taken by a drone," *IPSJ Transactions on Computer Vision and Applications*, vol. 11, no. 1, pp. 1–7, 2019.
- [16] D. Zhang, R. Watson, G. Dobie, C. MacLeod, A. Khan, and G. Pierce, "Quantifying impacts on remote photogrammetric inspection using unmanned aerial vehicles," *Engineering Structures*, p. 109940, 2020.
- [17] M. Sturzenegger and D. Stead, "Close-range terrestrial digital photogrammetry and terrestrial laser scanning for discontinuity characterization on rock cuts," *Engineering Geology*, vol. 106, no. 3, pp. 163–182, 2009.
- [18] M. Westoby, J. Brasington, N. Glasser, M. Hambrey, and J. Reynolds, "'structure-from-motion' photogrammetry: A low-cost, effective tool for geoscience applications," *Geomorphology*, vol. 179, pp. 300–314, 2012.

## References

- [19] M. Favalli, A. Fornaciai, I. Isola, S. Tarquini, and L. Nannipieri, "Multiview 3D reconstruction in geosciences," *Computers & Geosciences*, vol. 44, pp. 168–176, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0098300411003128>
- [20] C. W. VisualSFM, "A visual structure from motion system," 2011.
- [21] C. Sweeney, T. Hollerer, and M. Turk, "Theia: A fast and scalable structure-from-motion library," in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015, pp. 693–696.
- [22] Agisoft, "Metashape (photoscan)," <http://www.agisoft.com/>, 2010, accessed: 2020-06-15.
- [23] Bentley, "Contextcapture," <https://www.bentley.com/en/products/brands/contextcapture>, 2016, accessed: 2020-06-15.
- [24] CapturingReality, "Reality capture," <https://www.capturingreality.com/>, 2016, accessed: 2020-06-15.
- [25] S. Siebert and J. Teizer, "Mobile 3d mapping for surveying earthwork projects using an unmanned aerial vehicle (uav) system," *Automation in Construction*, vol. 41, pp. 1–14, 2014.
- [26] K. K. Han and M. Golparvar-Fard, "Appearance-based material classification for monitoring of operation-level construction progress using 4D BIM and site photologs," *Automation in Construction*, vol. 53, pp. 44–57, may 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0926580515000266>
- [27] S. Tutas, A. Braun, A. Borrmann, and U. Stilla, "Acquisition and consecutive registration of photogrammetric point clouds for construction progress monitoring using a 4d bim," *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 85, no. 1, pp. 3–15, 2017.
- [28] M. R. Jahanshahi, S. F. Masri, C. W. Padgett, and G. S. Sukhatme, "An innovative methodology for detection and quantification of cracks through incorporation of depth perception," *Machine vision and applications*, pp. 1–15, 2013.
- [29] K. Chaiyasarn, T.-K. Kim, F. Viola, R. Cipolla, and K. Soga, "Distortion-Free Image Mosaicing for Tunnel Inspection Based on Robust Cylindrical Surface Estimation through Structure from Motion," *Journal of Computing in Civil Engineering*, vol. 30, no. 3, p. 4015045, 2016.
- [30] S. Stent, R. Gherardi, B. Stenger, K. Soga, and R. Cipolla, "Visual change detection on tunnel linings," *Machine Vision and Applications*, vol. 27, no. 3, pp. 319–330, 2016.

## References

- [31] J. E. N. Masson and M. R. Petry, "Comparison of mesh generation algorithms for railroad reconstruction," in *Autonomous Robot Systems and Competitions (ICARSC), 2017 IEEE International Conference on*. IEEE, 2017, pp. 266–271.
- [32] A. Khaloo and D. Lattanzi, "Hierarchical Dense Structure-from-Motion Reconstructions for Infrastructure Condition Assessment," *Journal of Computing in Civil Engineering*, vol. 31, no. 1, p. 04016047, jan 2017. [Online]. Available: <http://ascelibrary.org/doi/10.1061/%28ASCE%29CP.1943-5487.0000616>
- [33] J. Schöning and G. Heidemann, "Evaluation of multi-view 3d reconstruction software," in *International conference on computer analysis of images and patterns*. Springer, 2015, pp. 450–461.
- [34] I. Nikolov and C. Madsen, "Benchmarking close-range structure from motion 3d reconstruction software under varying capturing conditions," in *Euro-Mediterranean Conference*. Springer, 2016, pp. 15–26.
- [35] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [36] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.
- [37] C. Wu, "Towards linear-time incremental structure from motion," in *3DTV-Conference, 2013 International Conference on*. IEEE, 2013, pp. 127–134.
- [38] M. Smith, J. Carrivick, and D. Quincey, "Structure from motion photogrammetry in physical geography," *Progress in Physical Geography*, vol. 40, no. 2, pp. 247–275, 2016.
- [39] N. D'Amico and T. Yu, "Accuracy analysis of point cloud modeling for evaluating concrete specimens," in *SPIE Smart Structures and Materials+ Nondestructive Evaluation and Health Monitoring*. International Society for Optics and Photonics, 2017, pp. 101 691D–101 691D.
- [40] N. Micheletti, J. H. Chandler, and S. N. Lane, "Investigating the geomorphological potential of freely available and accessible structure-from-motion photogrammetry using a smartphone," *Earth Surface Processes and Landforms*, vol. 40, no. 4, pp. 473–486, 2015. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/esp.3648>
- [41] M. W. Smith and D. Vericat, "From experimental plots to experimental landscapes: topography, erosion and deposition in sub-humid badlands

## References

- from Structure-from-Motion photogrammetry," *Earth Surface Processes and Landforms*, vol. 40, no. 12, pp. 1656–1671, sep 2015. [Online]. Available: <http://doi.wiley.com/10.1002/esp.3747>
- [42] N. Micheletti, J. H. Chandler, and S. N. Lane, "Structure from Motion (SfM) Photogrammetry," in *Geomorphological Techniques*, online ed. ed. British Society for Geomorphology Geomorphological Techniques, 2015, ch. 2.2, pp. 1–12. [Online]. Available: [http://geomorphology.org.uk/sites/default/files/chapters/2.2.2\\_{\\_}SfM.pdf](http://geomorphology.org.uk/sites/default/files/chapters/2.2.2_{_}SfM.pdf)
- [43] S. Harwin and A. Lucieer, "Assessing the accuracy of georeferenced point clouds produced via multi-view stereopsis from unmanned aerial vehicle (uav) imagery," *Remote Sensing*, vol. 4, no. 6, pp. 1573–1599, 2012. [Online]. Available: <http://www.mdpi.com/2072-4292/4/6/1573>
- [44] T. P. Kersten and M. Lindstaedt, "Image-based low-cost systems for automatic 3d recording and modelling of archaeological finds and objects," in *Euro-Mediterranean Conference*. Springer, 2012, pp. 1–10.
- [45] K. Thoeni, A. Giacomini, R. Murtagh, and E. Kniest, "A comparison of multi-view 3D reconstruction of a rock wall using several cameras and a Laser scanner," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 40, no. 5, pp. 573–580, 2014.
- [46] M. R. James and S. Robson, "Straightforward reconstruction of 3D surfaces and topography with a camera: Accuracy and geoscience application," *Journal of Geophysical Research: Earth Surface*, vol. 117, no. F3, 2012. [Online]. Available: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2011JF002289>
- [47] J. Goetz, A. Brenning, M. Marcer, and X. Bodin, "Modeling the precision of structure-from-motion multi-view stereo digital elevation models from repeated close-range aerial surveys," *Remote Sensing of Environment*, vol. 210, pp. 208–216, jun 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0034425718301068>
- [48] S. Tavani, A. Corradetti, and A. Billi, "High precision analysis of an embryonic extensional fault-related fold using 3D orthorectified virtual outcrops: The viewpoint importance in structural geology," *Journal of Structural Geology*, vol. 86, pp. 200–210, may 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0191814116300335><http://dx.doi.org/10.1016/j.jsg.2016.03.009>
- [49] A. Corradetti, K. McCaffrey, N. De Paola, and S. Tavani, "Evaluating roughness scaling properties of natural active fault surfaces by means of multi-view photogrammetry," *Tectonophysics*, vol. 717, pp. 599–606, 2017.

## References

- [50] H. Hansen, R. Hocken, and G. Tosello, "Replication of micro and nano surface geometries," *CIRP Annals - Manufacturing Technology*, vol. 60, no. 2, pp. 695–714, jan 2011. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0007850611002101>[#](http://www.sciencedirect.com/science/article/pii/S0007850611002101?via%3Dihub)
- [51] L. Nilsson and R. Ohlsson, "Accuracy of replica materials when measuring engineering surfaces," *International Journal of Machine Tools and Manufacture*, vol. 41, no. 13-14, pp. 2139–2145, oct 2001. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0890695501000803>
- [52] S. . Gasparin, H. N. Hansen, and G. Tosello, "Traceable surface characterization using replica moulding technology," pp. 310–315, 2011.
- [53] Y. C. Liu, C. Y. Ling, a. a. Malcolm, and Z. G. Dong, "Accuracy of replication for non-destructive surface finish measurement," in *Singapore International NDT Conference & Exhibition, 2011*, pp. 3–4. [Online]. Available: [http://www.ndt.net/article/SINCE2011/papers/30\\_{\\_}Liu.pdf](http://www.ndt.net/article/SINCE2011/papers/30_{_}Liu.pdf)
- [54] K. Walton, L. Fleming, M. Goodhand, R. Racasan, and W. Zeng, "High fidelity replication of surface texture and geometric form of a high aspect ratio aerodynamic test component," *Surface Topography: Metrology and Properties*, vol. 4, no. 2, p. 025003, apr 2016. [Online]. Available: <http://stacks.iop.org/2051-672X/4/i=2/a=025003?key=crossref.a9d0b86e0940cc23b86409417e80fc9f>
- [55] F. Baruffi, P. Parenti, F. Cacciatore, M. Annoni, and G. Tosello, "On the Application of Replica Molding Technology for the Indirect Measurement of Surface and Geometry of Micromilled Components," *Micromachines*, vol. 8, no. 6, p. 195, jun 2017. [Online]. Available: <http://www.mdpi.com/2072-666X/8/6/195>
- [56] C. Kumar, A. Palacios, V. A. Surapaneni, G. Bold, M. Thielen, E. Licht, T. E. Higham, T. Speck, and V. Le Hou  rou, "Replicating the complexity of natural surfaces: Technique validation and applications for biomimetics, ecology and evolution," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 377, no. 2138, 2019.
- [57] C. Bak, M. Gaunaa, A. S. Olsen, and E. K. Kruse, "What is the critical height of leading edge roughness for aerodynamics," in *Journal of Physics: Conference Series*, vol. 753, no. 2, 2016, p. 022023.

## References

- [58] J. Schöning and G. Heidemann, "Taxonomy of 3d sensors," *Argos*, vol. 3, no. P100, pp. 9–10, 2016.
- [59] I. Nikolov and C. Madsen, "Performance characterization of absolute scale computation for 3d structure from motion reconstruction," in *14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (Visigrapp 2019) International Conference on Computer Vision Theory and Applications*. SCITEPRESS Digital Library, 2019, pp. 884–891.
- [60] D. Girardeau-Montaut, "Cloudcompare-open source project," *Open-Source Project*, 2011.
- [61] "Struers 2010 repliset technical data sheet," <https://www.struers.com/en/Products/Materialographic-analysis/Materialographic-analysis-equipment/Replication-system>, accessed: 2020-03-06.
- [62] S. Jolivet, S. Mezghani, and M. El Mansori, "Multiscale analysis of replication technique efficiency for 3D roughness characterization of manufactured surfaces," *Surface Topography: Metrology and Properties*, vol. 4, no. 3, p. 035002, sep 2016. [Online]. Available: <http://stacks.iop.org/2051-672X/4/i=3/a=035002?key=crossref.1c039023d449863c30929509a2c5cf8a>
- [63] "Plu neox confocal microscope," <https://www.sensofar.com/sensofar-introduces-their-most-advanced-optical-3d-profiler-plu-neox/#>, accessed: 2020-03-06.
- [64] "Scanning probe image processor (spip)," <https://www.imagemet.com/products/spip/>, accessed: 2020-03-06.
- [65] P. de Groot and X. C. de Lega, "Interpreting interferometric height measurements using the instrument transfer function," in *Fringe 2005*. Springer Berlin Heidelberg, 2006, pp. 30–37.
- [66] J. Song, T. Vorburger, L. Ma, J. Libert, and S. Ballou, "A metric for the comparison of surface topographies of standard reference material (SRM) bullets and casings," in *Proceedings of the 20th Annual ASPE Meeting, ASPE 2005*, 2005.
- [67] "Iso 25178-3(2012) geometrical product specifications (gps) – surface texture: Areal – part 3: Specification operators," <https://www.iso.org/standard/42895.html>, accessed: 2020-03-05.
- [68] "Iso 25178 geometrical product specifications (gps) – surface texture: Areal – part 2: Terms, definitions and surface texture parameters," <https://www.iso.org/standard/42785.html>, accessed: 2020-03-05.

## References

- [69] J. S. Nielsen and J. D. Sørensen, "Bayesian estimation of remaining useful life for wind turbine blades," *Energies*, vol. 10, no. 5, p. 664, 2017.

## References

# Paper I

## Preliminary Study on the use of Off-the-Shelf VR Controllers for Vibrotactile Differentiation of Levels of Roughness on Meshes

Ivan Nikolov, Jens Stokholm Høngaard, Martin Kraus, Claus  
Madsen

The paper has been published in the  
*Proceedings of 15th International Joint Conference on Computer Vision, Imaging  
and Computer Graphics Theory and Applications* pp. 334-340, 2020.

© 2020 SCITEPRESS Digital Library  
*The layout has been revised.*

## Abstract

*With the introduction of new specialized hardware, Virtual Reality (VR) has gained more and more popularity in recent years. VR is particularly immersive if suitable auditory and haptic feedback is provided to users. Many proposed forms of haptic feedback require custom hardware components that are often bulky, costly, and/or require lengthy setup times. We explored the possibility of using the built-in vibrotactile feedback of HTC Vive controllers to simulate the sensation of interacting with surfaces with varying degrees of roughness. We conducted initial testing on the proposed system, which shows promising results as users could accurately and within short time discern the amount of roughness of 3D models based on the vibrotactile feedback alone.*

## I.1 Introduction

In recent years there has been a steady rise of the number and quality of VR solutions. All these systems aim to immerse users by using a combination of visual and audio modalities together with a sense of presence in the VR environment, which is achieved by internal or external head and hand tracking. For interacting with the VR environment, the state-of-the-art solutions usually rely on controllers. However, the reliance on controllers and the impossibility to touch and feel models in the 3D environment have hampered immersion. Virtual reality applications in many areas can benefit from the introduction of haptics, such as phantom limb pain [1] and stroke rehabilitation [2, 3], interactions for blind users [4], data visualization [5], cultural heritage [6], etc. Normally this is done through the use of custom hardware, which makes reproducibility hard and expensive.

In this paper we present an initial study on the use of the built-in vibration motors in the HTC Vive [7] controllers for detecting and differentiating different levels of roughness on meshes in VR. We tested our solution on fifteen participants of varying VR skill levels, using high detail 3D reconstructions of real world objects to achieve natural interactions. The participants did not see the real roughness of the object, but could only perceive it through the tactile sensation provided by the vibrotactile feedback of the controller. All users, independent of their skill level, managed to correctly distinguish the different levels of roughness of the 3D objects in a short amount of time. Thus, the research in this paper serves as a proof of concept that different levels of roughness can be successfully communicated through VR controllers without any additional hardware.

## I.2 State of the Art

Haptic feedback has two main parts - kinesthetic and tactile feedback. Kinesthetic feedback uses the feeling coming from a person's muscles and tenders to distinguish the object that is being touched, grabbed, or held. Tactile feedback comes from the feeling of the skin sensors on the fingers and palms when an object is touched and can convey the shape, texture, and roughness. Introduction of haptic feedback to VR solutions is a non-trivial problem. This paper is focused solely on tactile feedback.

Haptic interfaces can be divided into passive and active. Both types can be useful for different cases in VR. Passive ones rely on the shape of the controller and try to mimic real life objects or surfaces. Examples of these can be seen in the work of [8, 9]. Active haptic feedback controller rely on moving parts, actuators and sensors, to dynamically mimic changes in the environment or the virtual objects. Examples of active haptics can be found in [10–12]. Active haptic controllers are of bigger interest to the current study.

Another possibility of active haptics is the introduction of custom tactile controllers, as seen in the work by [13–15] or sensors directly attached to the users' fingertips [16, 17]. These controllers rely on a combination of actuators, inertial measurement units and electromagnetic coils to create a very precise sense of touch, but they require custom hardware, are normally quite bulky and are not readily available for the general public. Work is also done on directly using the controllers coming with the VR systems. Some research focuses on augmenting the controllers with additional functionality [18–20], while others rely directly on controllers' vibration [21, 22].

We base the study in this paper on the idea that controller vibration can give an active haptics idea of the surface of 3D objects in VR and help users differentiate levels of roughness.

## I.3 Methodology

Our proposed approach follows the research using integrated controllers, as this makes it easier to replicate and test, as well as simpler to introduce and explain to users. This means that the sensation of tactile feedback needs to be simulated to the user, so the proper information is understood. Our hypothesis is that the built-in vibrotactile features in the HTC Vive controllers can achieve this sensation, but only if the provided vibration motors are carefully controlled. In this way, VR interactions with 3D models can be achieved that are relatively close to touching a surface with a hand-held stylus in the real world.

The implementation uses Unity with the SteamVR plugin [23]. The Steam VR API exposes three parameters for modulating the vibration of the VR

### I.3. Methodology

controllers: amplitude, frequency and duration of the vibration. Each of these parameters has set value constraints:

- Amplitude can take floating values between [0..1]
- Frequency accepts floating values between [0..350]
- Duration of vibration accepts floating values for the amount of seconds, with a lower bound value determined by the hardware limitations of the vibration motor

A limitation of the vibration feature is that the motors can not run all the time, therefore we have set a heuristic minimum distance of 5 mm between sampling mesh surface points before which the controller's motors are not started. This will ensure that there are pauses between the repeated activations of the haptic motor and will limited the produced vibration noise. Additionally to mitigate the possibility of noise we sample the surface of the object once every 0.1 seconds.



**Fig. I.1:** Rendering of our virtual VR stylus connected to the VR controller. The virtual stylus is used to virtually "touch" objects. The stylus is only seen in VR and does not exist on the real controller. The red rays cast from the stylus are shown for easier explanation of method and are not visible, when using the application.

Our system detects the underlying mesh roughness by calculating the angle between two sampled surface normals. To help with directing the users, in VR a 3D model of a stylus is placed on top of the Vive controller as seen in Figure I.1.

Rays are cast from each vertex point of the mesh of the stylus in the direction of their surface normal vectors. The intersections of these rays with other surfaces determine the contact point between the stylus and another surface. The maximum allowed distance of intersection points is dependent

on the shape of the stylus, with additional length to ensure contact at the various possible orientations. All contact points are checked starting from the tip of the stylus and going down. When two contact points are sampled, the angle between their normals is calculated and analysed. This angle can be between  $[0, \pi]$ , as seen in the work of [24], as smooth movement on the stylus on the surface is assumed.

As this is a pilot study, it was decided that the most straightforward approach is to lock the duration and leave the amplitude as the only actively adjustable quality in the experiments. Thus, the underlying surfaces can be approximated without the need to map the physical surface profile to vibration frequency. This introduces the problem that smaller surface details cannot be communicated through the vibration. To mitigate that and after analysing the difference between the normals, we simplify the underlying surface roughness to a binary classification for the vibration controller:

- Surface patches with large angle between the normals result in vibration with high amplitude, i.e., momentary, tactile "bumps" – approximating very rough areas
- Surface patches with small angle between the normal result in low-amplitude, continuous vibration – approximating ambient roughness.

The two cases are distinguished by considering the calculated angle difference: if it is less than 6 degrees, then it is a low-amplitude vibration patch; if it is greater than 6 degrees, then it is hard vibration case. This threshold was selected heuristically after multiple trials as a believable approximation of the underlying tested surfaces. Thus our system has a dynamic component in changing the amplitude and passive component in changing between levels of predetermined frequency. The values of the frequency levels are selected after a number of internal trials:

- In case of a large normal angle the vibration duration is set to 0.075 seconds and the frequency to 16 Hz.
- In case of a small normal angle the vibration duration is set to 0.025 seconds with a frequency of 344 Hz.

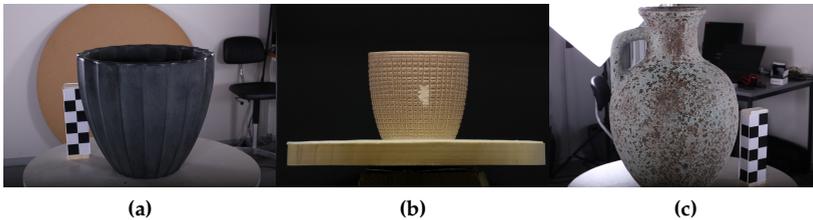
The amplitude in both cases is dynamically modulated depending on the difference between the normals and the distance between the sampled points. The distance between samples is used to change the amplitude, to better approximate the feeling of dragging the stylus on a real surface. If we consider that the motion on the surface is continuous, the larger the distance between samples, taken at equal time steps, the faster the stylus is moving. Our hypothesis is that faster movement has the tendency to "smooth out" the feeling of a rougher surfaces.

The vibrations are only sent if the user’s finger is on the trackpad since this is the part of the Vive controller where the vibrotactile feedback is felt most distinctly due to the position of the vibration motor.

### I.4 Experiment and Results

To test how much information about the object’s roughness our proposed solution can offer users, we designed an experiment, which relied solely on the tactile information.

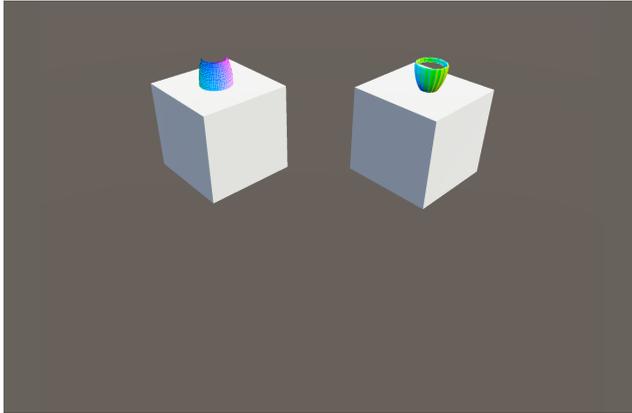
#### I.4.1 Experiment Setup



**Fig. I.2:** Three objects used in the experimental evaluation. The first two vases I.2a and I.2b are used in the initial training phase, while the third vase is used in the testing part.

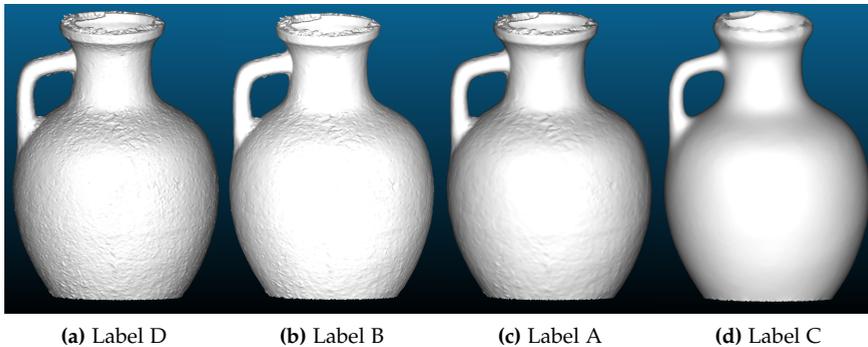
Three real world vases were selected and digitized using Structure from Motion (SfM) reconstruction, through the commercial software Metashape [25]. The vases were selected because of their roughness profiles. The two vases in Figure I.2a and Figure I.2b have a simplified roughness profile of a wave and a checkerboard pattern, which was useful for the training phase of the experiment, where users were familiarized with the setup and left to explore it, until they felt comfortable with it. These patterns provided an easy way to understand the relation between the visual appearance in VR and the tactile sensation that the controller vibration provides when interacting with the objects. A view from the initial training part can be seen in Figure I.3. The real world objects were selected, to give participants an object that both has small scale roughness, but also large scale surface shape. Our hypothesis is that this will make distinguishing the different levels of roughness harder and will limit the possible effects from users learning the roughness from for example a planar shape.

The third vase, seen in Figure I.2c, was used as a basis for the experiment. Three copies of the reconstructed 3D mesh were made and each was smoothed. Three degrees of smoothing were utilized which were generated by Laplacian smoothing in Meshlab [26]. The original reconstruction and the



**Fig. I.3:** View from the training area, where the users could try out and test the tactile feedback on the two simple surface objects.

three smoothed copies can be seen in Figure I.4. Because we were not checking if users can precisely measure roughness, but if users can distinguish and order different degrees of roughness, the degree of smoothness were chosen heuristically by experimenting with different configurations. To be sure that the vase and the three smoothed copies follow a "smoothness progression", a patch was taken from each of them and the root mean square height  $S_q$  was calculated from each one [27]. The least rough vase was almost completely smoothed, to provide an almost blank slate compared to the other three.



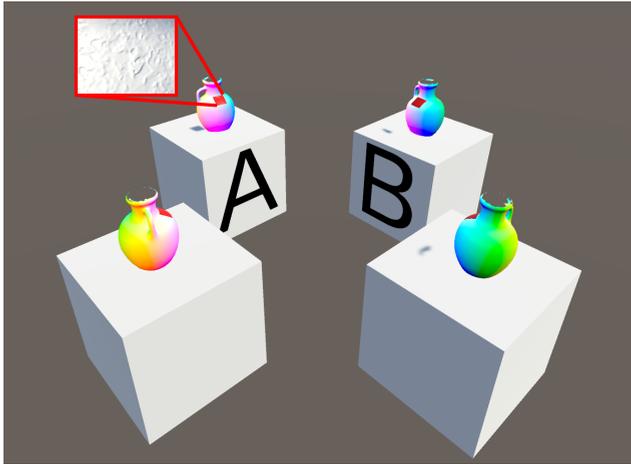
**Fig. I.4:** The vase used for the testing phase (a), together with the other three progressively smoother copies (b) to (d). The labels from A to D were set as seen in Table I.1

The four vases were set on pedestals in VR with letters A, B, C, and D. The roughness levels, together with the set label letters and the calculated  $S_q$  are shown in Table I.1. The letters are purposely randomly assigned depending

#### I.4. Experiment and Results

**Table I.1:** The roughness levels of each of the four objects, together with the labels denoting them and the calculated root mean square height  $S_q$ . The letters are given at random to the different roughness levels and are used when testing

Roughness	$S_q$ [mm]	Label
Most	0.5569	D
Second	0.5114	A
Third	0.4784	B
Least	0.0304	C



**Fig. I.5:** View from the testing area, with the four identically looking vases. Each of the vases has the collider of a object with different level of roughness. The objects are labeled and a red patch is selected on them, to help directing the attention of the users. A close-up of the underlying patch roughness is shown for easier understanding and is not visible for the users. The objects are rotated between users.

on the roughness level. The rendering of their meshes was deactivated with only the colliders left. On top of each of them a completely smooth model was rendered, so users had no way of visually seeing the real roughness. The testing setup can be seen in Figure I.5. Between each user test the positions of the vases were randomly rotated, but the combination between letter and roughness was not changed. We rotate the vases to avoid directional bias from users, when checking which object they interact most with. This bias can manifest in right or left handed participants going always to the object on their left or right, which without the rotation can always be the same. To help directing the attention of the users, a specific patch of all the vases was selected, where the roughness is particularly pronounced and colored red. A close-up of underlying roughness of the patch is shown in Figure I.5, while the users just saw a smooth red surface.

## I.4.2 Participants and Captured Data

Fifteen participants tested the system by using the experimental setup. The users had an age between 23 and 35 years and varying degrees of proficiency using VR. Each user was left to first explore the training part of the experimental VR setup, while the facilitator explained to them how to use the system. Once the user was comfortable, they were teleported to the testing part, where they were asked to try to order the four identically looking vases depending on the perceived roughness profile, when they interacted with them. The users had unlimited time and were instructed to use the specified red patch on the vases if they had a hard time distinguishing the objects. Once the users were ready, they gave their idea of the ordering of the vases. The time between the start of the experiment and the end was taken, as well as the amount of times the user had interacted with each of the four vases.

## I.4.3 Results and Discussion

All fifteen participants could successfully order the objects from roughest to smoothest. The average completion time was 124.8 seconds, with a standard deviation of 68.3 seconds. The large standard deviation was caused by three participants, who took more than 200 seconds to complete the experiment. All three of the participants had little or no experience in using VR, thus, their slower completion time can be attributed to some extent to their inexperience. The completion times for all users, depending on their proficiency, can be seen in Figure I.6. Here it can be seen that some of the people with no proficiency took a lot of time, due to not being fully comfortable using a VR controller and overall a lot more variation is seen in their times. People with high proficiency have a lower spread and do not require a lot of time to distinguish between roughness levels successfully.

The number of times users interacted with each of the four objects can give an overview, which object was the most difficult to categorize. The results for each user, depending on their proficiency level, for each of the objects can be seen in Figure I.7. These results reflect the completion times, discussed above, with some deviations, showing that some users were interacting with the objects more, while other were more passive. Again users with no proficiency required more interactions and focus more on the smoothest object C.

Table I.2 expands more on the captured results. The values in *Most Interactions* denote the number of times each of the four objects has been interacted with the most by the users. On the other hand, *Least Interactions* denotes the number of times each object has been interacted with the least amount of times. The table shows that the object with the most roughness *D*, has never been interacted the most times and has been interacted the least amount of

## I.5. Conclusion and Future Work

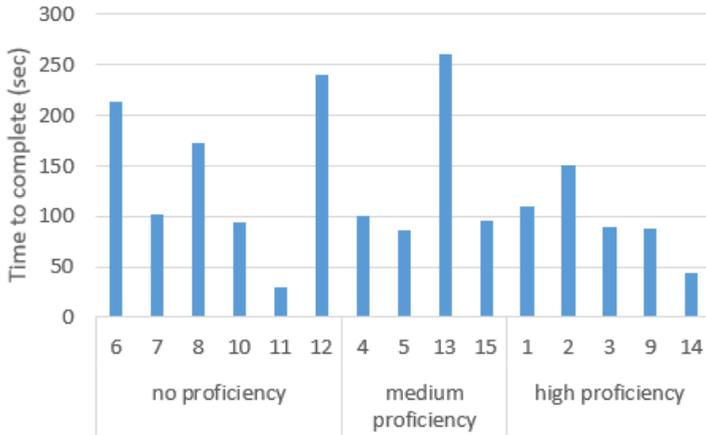


Fig. I.6: Completion times for each user, grouped by the three VR proficiency levels.

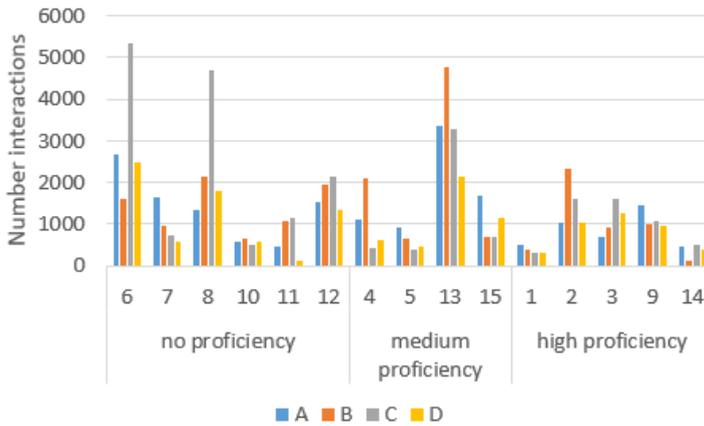
time by most users. This shows that people generally really easily decided how rough it was and did not need more interaction. On the other hand, the smoothest object C is both the most interacted object by the most people, as well as the second least interacted object. This points to the fact that the order in which users interacted with the objects is important for discerning their correct roughness. As expected, starting with the roughest object is most beneficial. Finally, the two in-between objects A and B, had approximately the same level of difficulty, but once users made up their minds on the most and least rough, they could decide on the in-between ones easier. On top of that some participants commented on that they could hear the haptic motors spin, those who did comment on it were told to try and ignore it. But it is unclear if it had an impact on the results.

Table I.2: The most and least interacted with labeled objects. The results are created from the fifteen participants.

	A	B	C	D
Most Interactions	5	4	6	0
Least Interactions	3	3	4	5

## I.5 Conclusion and Future Work

Our experiment demonstrated that information about the surface roughness of 3D objects can be communicated through the use of tactile sensation achieved by the built-in vibration capabilities of HTC Vive controllers. With the help of a virtual VR stylus, users can use the same natural interactions



**Fig. I.7:** Number of times users interacted with each of the four objects, grouped by the three VR proficiency levels. Each time users touched the objects, this is counted as an interaction.

as in real life to "feel" the surface of an object. The preliminary experiment demonstrated that users can order objects by their perceived vibration surface roughness without visual cues. The test showed that users could relatively fast decide which is the roughest of multiple, visually identical objects, as well as the smoothest and order them always correctly.

There are some limitations of this preliminary study. The limited scope of the test and the limited number of participants resulted in results which are too homogeneous and do not show enough variation to further improve the system. To address these shortcomings, additional experiments are planned. In particular, we want to investigate the influence of amplitude and frequency of the vibrotactile feedback on the perceived roughness. Another experiment could explore how much of an impact the sound of the controller has on the haptic feeling as some of the test participants mentioned that they could hear different noises from the motors in the controllers.

## Acknowledgements

We would like to thank the participants in the experiment for their time and the anonymous reviewers for providing helpful feedback and improving the content of the paper.

## References

- [1] B. Henriksen, R. Nielsen, M. Kraus, and B. Geng, "A virtual reality system for treatment of phantom limb pain using game training and tactile feedback," in *Proceedings of the Virtual Reality International Conference-Laval Virtual 2017*. ACM, 2017, p. 13.
- [2] M. R. Afzal, H.-Y. Byun, M.-K. Oh, and J. Yoon, "Effects of kinesthetic haptic feedback on standing stability of young healthy subjects and stroke patients," *Journal of neuroengineering and rehabilitation*, vol. 12, no. 1, p. 27, 2015.
- [3] M. F. Levin, E. C. Magdalon, S. M. Michaelsen, and A. A. Quevedo, "Quality of grasping and the role of haptics in a 3-d immersive virtual reality environment in individuals with stroke," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 6, pp. 1047–1055, 2015.
- [4] O. Schneider, J. Shigeyama, R. Kovacs, T. J. Roumen, S. Marwecki, N. Boeckhoff, D. A. Gloeckner, J. Bounama, and P. Baudisch, "Dualpanto: A haptic device that enables blind users to continuously interact with virtual worlds," in *The 31st Annual ACM Symposium on User Interface Software and Technology*. ACM, 2018, pp. 877–887.
- [5] R. Englund, K. L. Palmerius, I. Hotz, and A. Ynnerman, "Touching data: Enhancing visual exploration of flow data with haptics," *Computing in Science & Engineering*, vol. 20, no. 3, pp. 89–100, 2018.
- [6] M. H. Jamil, P. S. Annor, J. Sharfman, R. Parthesius, I. Garachon, and M. Eid, "The role of haptics in digital archaeology and heritage recording processes," in *2018 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE)*. IEEE, 2018, pp. 1–6.
- [7] HTC, "Vive," <https://www.vive.com/eu/>, 2016, accessed: 2019-08-15.
- [8] L.-P. Cheng, L. Chang, S. Marwecki, and P. Baudisch, "iturk: Turning passive haptics into active haptics by making users reconfigure props in virtual reality," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 2018, p. 89.
- [9] A. Zenner and A. Krüger, "Shifty: A weight-shifting dynamic passive haptic proxy to enhance object perception in virtual reality," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 4, pp. 1285–1294, 2017.

## References

- [10] J. Ryu, J. Jung, S. Kim, and S. Choi, "Perceptually transparent vibration rendering using a vibration motor for haptic interaction," in *RO-MAN 2007-The 16th IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 2007, pp. 310–315.
- [11] S. Scheggi, L. Meli, C. Pacchierotti, and D. Prattichizzo, "Touch the virtual reality: using the leap motion controller for hand tracking and wearable tactile devices for immersive haptic rendering," in *ACM SIGGRAPH 2015 Posters*. ACM, 2015, p. 31.
- [12] E. Whitmire, H. Benko, C. Holz, E. Ofek, and M. Sinclair, "Haptic revolver: Touch, shear, texture, and shape rendering on a reconfigurable virtual reality controller," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 2018, p. 86.
- [13] I. Choi, E. Ofek, H. Benko, M. Sinclair, and C. Holz, "Claw: A multifunctional handheld haptic controller for grasping, touching, and triggering in virtual reality," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 2018, p. 654.
- [14] H. Benko, C. Holz, M. Sinclair, and E. Ofek, "Normaltouch and texturetouch: High-fidelity 3d haptic shape rendering on handheld virtual reality controllers," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 2016, pp. 717–728.
- [15] H. Culbertson, J. M. Walker, M. Raitor, and A. M. Okamura, "Waves: a wearable asymmetric vibration excitation system for presenting three-dimensional translation and rotation cues," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 2017, pp. 4972–4982.
- [16] S. B. Schorr and A. M. Okamura, "Fingertip tactile devices for virtual object manipulation and exploration," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 2017, pp. 3115–3119.
- [17] V. Yem, R. Okazaki, and H. Kajimoto, "Fingar: combination of electrical and mechanical stimulation for high-fidelity tactile presentation," in *ACM SIGGRAPH 2016 Emerging Technologies*. ACM, 2016, p. 7.
- [18] P.-H. Han, Y.-S. Chen, K.-T. Yang, W.-S. Chuan, Y.-T. Chang, T.-M. Yang, J.-Y. Lin, K.-C. Lee, C.-E. Hsieh, L.-C. Lee *et al.*, "Boes: attachable haptics bits on gaming controller for designing interactive gameplay," in *SIGGRAPH Asia 2017 VR Showcase*. ACM, 2017, p. 3.
- [19] Y.-S. Chen, P.-H. Han, J.-C. Hsiao, K.-C. Lee, C.-E. Hsieh, K.-Y. Lu, C.-H. Chou, and Y.-P. Hung, "Soes: Attachable augmented haptic on gaming

## References

- controller for immersive interaction,” in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 2016, pp. 71–72.
- [20] A. Ryge, L. Thomsen, T. Berthelsen, J. S. Hvass, L. Koreska, C. Vollmers, N. C. Nilsson, R. Nordahl, and S. Serafin, “Effect on high versus low fidelity haptic feedback in a virtual reality baseball simulation,” in *2017 IEEE Virtual Reality (VR)*. IEEE, 2017, pp. 365–366.
- [21] J. Kreimeier and T. Götzelmann, “Feelvr: Haptic exploration of virtual objects,” in *Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference*. ACM, 2018, pp. 122–125.
- [22] P. W. Brasen, M. Christoffersen, and M. Kraus, “Effects of vibrotactile feedback in commercial virtual reality systems,” in *Interactivity, Game Creation, Design, Learning, and Innovation*. Springer, 2018, pp. 219–224.
- [23] Valve, “Steam vr,” <https://github.com/ValveSoftware/steamvr>, 2015, accessed: 2019-08-20.
- [24] Y. Ioannou, B. Taati, R. Harrap, and M. Greenspan, “Difference of normals as a multi-scale operator in unorganized point clouds,” in *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*. IEEE, 2012, pp. 501–508.
- [25] Agisoft, “Metashape (photoscan),” <http://www.agisoft.com/>, 2010, accessed: 2020-06-15.
- [26] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, “MeshLab: an Open-Source Mesh Processing Tool,” in *Eurographics Italian Chapter Conference*, V. Scarano, R. D. Chiara, and U. Erra, Eds. The Eurographics Association, 2008.
- [27] K. Carneiro, *Scanning Tunnelling Microscopy Methods for Roughness and Micro Hardness Measurements*. Directorate-General XIII, 1995.

## References

# Paper J

## Quantifying Wind Turbine Blade Surface Roughness Using Sandpaper Grit Sizes

Ivan Nikolov and Claus Madsen

The paper is under review at  
*16th International Joint Conference on Computer Vision, Imaging and Computer  
Graphics Theory and Applications*

*The layout has been revised.*

## Abstract

*Surface inspection of wind turbine blades is a necessary step, to ensure longevity and sustained high energy output. The detection of accumulation of damages and increased surface roughness of in-use blades, is one of the main objectives of inspections in the wind energy industry. Creating 3D scans of the leading edges of blade surfaces has been more and more used for capturing the roughness profile of blades. An important part in analysing these surface 3D scans is the standardization of the captured data across different blade surfaces, types and sizes. In this paper we propose using sandpaper grit sizes to quantify the surface characteristics of captured blade 3D scans. Sandpaper has been widely used for approximating different levels of blade surface roughness and its standardized nature can be used to easily describe and compare blade surfaces. We 3D reconstruct a number of different sandpaper grit sizes - from coarser P40 to a finer P180. We extract a number of 3D surface features from them and use them to train a random forest classification method. This method is then used to segment the surfaces of wind turbine blades in areas of different surface roughness. We test our proposed solution on a variety of blade surfaces - from smooth to course and damaged and show that it manages to classify them depending on their roughness.*

## J.1 Introduction

Surface inspection is a required part of ensuring the proper working condition of machinery and infrastructure in industries like agriculture [1], transportation [2], manufacturing [3], energy production [4] among other. An industry that is particularly susceptible to the effects of infrastructure damages and degradation is the wind energy production industry. For achieving maximum wind turbine performance, blades need to be inspected regularly and potential damages caused by weather erosion, animals and imperfections in the manufacturing process [5–7], need to be detected as soon as possible. It has been shown that the presence of even small surface roughness deviations and damages can cause 2% and 5% loss in energy production [8], with numbers as high as 25% for larger damages and surface imperfections [9].

Wind turbine blade inspection is normally focused on the leading edge of the blades, as it is shown that damages there affect performance the most [10]. Inspection can be done manually by experts on site or in a laboratory setting through contact measurements and microscopy analysis [11, 12]. Another possible way that combines the precision and thoroughness of a structured laboratory inspection, without the need for stopping production or disassembling is by using machine vision methods [13, 14]. This normally requires capturing images of the surface, which are used for detecting potential damages and imperfection. 2D Images can be used for training vision based

systems for detection of damaged areas and capturing measurement on the severity of the damages [15, 16]. These algorithms require a lot of image data of both clean and damaged surfaces of wind turbine blades, which is information that is not easily accessible. These systems also cannot normally quantify the micro surface roughness of the blade, as they lack depth information. This depth information can be captured by reconstructing the 3D surface, using a techniques like Structure from Motion [17–19]. This 3D data can then be used to extract information, about the surface quality.

To quantify this information and be able to compare it between blade surfaces, a standardized model of measurement of surface roughness is required. Profile roughness metrics part of the ISO 4287 [20] standard are used for estimation of the details of line scans of surfaces. Examples of these metrics are the arithmetic mean deviation of the assessed profile  $R_a$ , the root mean square deviation of the profile  $R_q$ , the maximum peak  $R_p$  and valley  $R_v$  depth of the profile, etc. These profile metrics can be extended to a 3D area, by using the ISO 25178 [21] standard, a plane is fitted to the surface to patches and the metrics are calculated from their values. In this paper we want to extend these metrics representation, by classifying the surface based on sandpaper roughness that best describes it. Sandpaper with different sizes of grit has been widely used in the literature for modeling wind turbine blade roughness in testing scenarios in wind tunnels [22–24]. It has been shown that surfaces with attached sandpaper patches exhibit the same behaviour as if they had the same roughness profile. By classifying wind turbine blades, into different sandpaper roughness profiles, their degradation can be more easily communicated and compared between blades. In addition, sandpaper surface grits are standardized [25] and easily accessible, making reproduction of results easy and straightforward. Finally, 3D data of sandpaper surfaces can be much easily obtained and used for training of a roughness detection methods, than real wind turbine blade surface data with different levels of surface damages.

## J.2 State of the Art

Wind turbine blade analysis can be used to calculate the predicted energy production [26] and the utilization coefficients of turbines [27]. It can be also used for introducing a more granular control on the flow control and aerodynamic properties of blades [28, 29].

Capturing of 3D data from the surface of wind turbine blades is a widely researched topic. Two main approaches to capturing 3D can be seen in the state of the art - for on-site inspections [30], when the wind turbine has been just stopped and off-line laboratory inspections, where decommissioned blades are normally inspected [12]. The first type of inspection is

performed on a more regular basis and aims to keep the blades in as close as possible optimal conditions, while the second type is focus on understanding why and how a blade failed or it was decided that the damages are too severe. For on-site inspections normally use cameras and 3D sensors like LiDARs are mounted on unmanned aerial vehicles [17, 31]. These aim to produce less high detailed reconstructions, which could under optimal capturing conditions capture enough information to give an estimate of the current condition of the blades. Laboratory inspections normally use more detailed surface analysis technique, employing electronic microscopes or surface probes [32, 33]. These methods can capture very high resolution sub-millimeter accuracy reconstructions of the blade surfaces.

In this paper we try to classify wind turbine blade surfaces using sandpapers with different grit sizes. As sandpapers have known surface properties, we can then infer the same information about the blade surfaces. In addition, sandpaper data is widely used for simulating how different surface roughness affects the aerodynamic characteristics of wind turbine blades and their energy production potential. We base our research on traditional supervised learning methods, using hand crafted features like the ones presented in the work of [34–37], for use on large scale LiDAR point cloud segmentation, focusing on extracting geometrical features, which describe the underlying surface in a robust way. Our proposed solution uses 3D sandpaper data to train a Random Forest classifier, which is later used to segment wind turbine blade surfaces into areas best described by the different sandpaper grit sizes. This information can be later used to model test setups or to better evaluate how the roughness would affect the blade, based on sandpaper wind tunnel tests. We show that the proposed approach gives promising results.

## J.3 Methodology

### J.3.1 SfM Overview

As both the sandpaper patches and the testing data is captured using Structure from Motion (SfM), we will first give an overview of the method, for easier reproducibility. SfM is a part of the multi-view stereo algorithms, that uses only 2D image data from different positions and view directions to reconstruct the full 3D surface of an object [38]. A number of images with a certain overlap are used as input to the algorithm. Features are extracted from each image and matched between images. This can be done using algorithms like SIFT [39] or ORB [40]. These matched features, together with information about the intrinsic parameters of the camera used to take the images, are used to iteratively triangulate the camera positions and create a sparse point cloud of the scanned object. The sparse point cloud is then

refined using bundle adjustment [41] and a dense point cloud is interpolated by then known 3D sparse positions. Finally, a mesh can be created from the dense point cloud, together with textures from the reprojected image information.

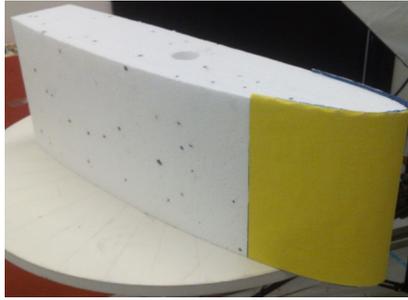
### J.3.2 Sandpaper 3D Reconstruction

**Table J.1:** Sandpaper grit size and the nominal average particle diameter in *mm* [42]. The notation of each grit represents the size of the particles in it.

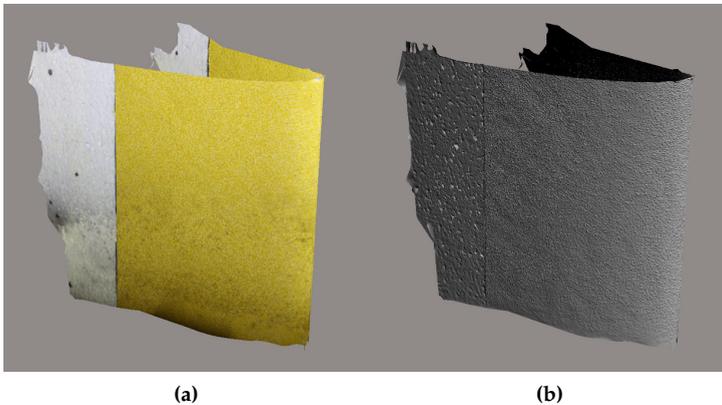
Grit size	Nom. av. diam. ( <i>mm</i> )
P40	0.425
P60	0.269
P80	0.201
P100	0.162
P120	0.125
P180	0.082

A number of sandpaper grit sizes are chosen for developing the training examples and given in Table J.1, together with their average particle size. This standard is proposed by [42] and widely used. The "P" notation of the grit sizes is inversely related to the coarseness of the sandpaper material and represents the size of the particles, embedded in the material. These grit sizes are chosen, because they are widely used for blade roughness approximation [24] and because the nominal average diameter of their grit structure is representative of blade roughness values, at which energy production loss starts to be observed [10]. Structure from Motion is chosen for 3D reconstruction of the sandpaper surfaces, as it provides high degree of accuracy, without the need for specialized sensors or hardware. To more closely mimic how roughness on a wind turbine blade edge would behave, the sandpaper patches are mounted on blade replicas made from styrofoam. The replicas are modeled after the NACA 633418 blade and can be seen in Figure J.1.

For 3D reconstructing the blade replicas a Canon 5Ds camera is used. As proposed by [43], a semi-circular pattern with a 1.5m radius is used and 18 images are taken in three different heights of the leading edge with the sandpaper attached. The thus created 54 images are then used as an input to Metashape [44], a commercial SfM solution, shown to produce high detailed and robust reconstructions. The large amount of captured images is chosen, to ensure that enough detail is captured for each grit size, so it is correctly represented and minimizing the possibility of surface imperfections [45]. An example of the resultant point clouds for grit size P40, with and without a texture can be seen in Figure J.2.



**Fig. J.1:** Example of styrofoam blade replica, with sandpaper installed on the leading edge. The shown sandpaper has a grit size of P40



**Fig. J.2:** Example of the leading edge reconstructions with sandpaper attached to them - with and without a texture

#### J.3.3 Sandpaper Feature Extraction

We extract a number of features from the sandpaper point clouds, as presented in the work by [46]. To calculate these features first the local neighbourhood around each point needs to be extracted. A faster binary searching for the distances between all points is done using a KDtree [47], using the implementation from [48]. In their work [46] and [35] test features extracted from different types of neighbourhood selection for each point - single scale neighbourhood of fixed size, single scale neighbourhood of changing size, combination of multiple scales of neighbourhoods and even a combination of multiple scales of neighbourhoods of different shape. From the results of their work it can be seen that a combination of multiple neighbourhoods gave the best results, of extracting descriptive features. We choose to this ap-

proach, but focus on neighbourhoods of the same shape, as the reason they used ones with different shapes was that the classes they were trying to detect were considerably varying in shapes and sizes. This is in sharp contrast to the uniform sandpaper roughness structures, we are working with. We choose the neighbourhood scales, in a way that they can encompass the average particle sizes of the selected sandpaper grits as seen in Table J.1. We choose spherical neighbourhoods with radii in the interval  $[1 \dots 0.1]$  mm and a delta change of 0.1 mm. This creates 10 progressively smaller scales of neighbourhoods. If not enough points are present in the neighbourhood for calculating the features, it is zeroed out.

Three types of features are selected, as described in [46] and [35], [37] - fundamental geometrical properties of the point clouds, local shape covariance features and local statistical shape distribution features. These features are calculated for each neighbourhood scale and combined. As described by [46], the fundamental geometrical properties are as follow:

- Local point density of the neighbourhood around a given point (Figure J.3a),
- The farthest distance between points in the neighbourhood (Figure J.3b),
- The maximum height of difference between points in the neighbourhood, where height is expressed in the direction of the average normal of the neighbourhood (Figure J.3c),
- The standard deviation of the height differences between points in the neighbourhood, where again the height is expressed in the direction of the average normal (Figure J.3d).

A visualization of these features for a neighbourhood size of  $1\text{mm}$  on the P40 sandpaper are shown in Figure J.3, where just a small area of the whole sandpaper leading edge is visualized.

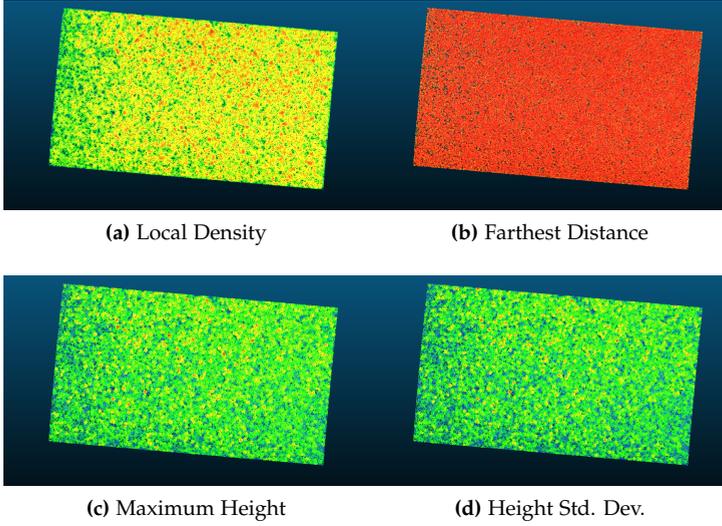
The local shape features [46] and [35] are based on the use of covariance features [49] and require the calculation of the eigenvalues  $\lambda_1, \lambda_2, \lambda_3$  of each neighbourhood scale. These features are as follows:

$$\text{Linearity: } L_\lambda = \frac{\lambda_1 - \lambda_2}{\lambda_1} \quad (\text{J.1})$$

$$\text{Planarity: } P_\lambda = \frac{\lambda_2 - \lambda_3}{\lambda_1} \quad (\text{J.2})$$

$$\text{Sphericity: } S_\lambda = \frac{\lambda_3}{\lambda_1} \quad (\text{J.3})$$

### J.3. Methodology



**Fig. J.3:** Example local fundamental geometrical properties, extracted from the P40 sandpaper grit size. Just a small part of the sandpaper leading edge is shown for easier visualization.

$$\text{Omnivariance: } O_\lambda = \sqrt[3]{\prod_{i=1}^3 \lambda_i} \quad (\text{J.4})$$

$$\text{Anisotropy: } A_\lambda = \frac{\lambda_1 - \lambda_3}{\lambda_1} \quad (\text{J.5})$$

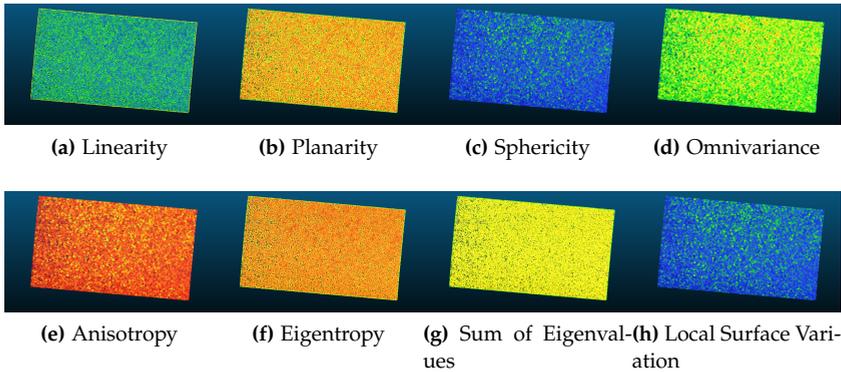
$$\text{Eigenentropy: } E_\lambda = - \sum_{i=1}^3 \lambda_i \ln \lambda_i \quad (\text{J.6})$$

$$\text{Sum of Eigenvalues: } \Sigma_\lambda = \sum_{i=1}^3 \lambda_i \quad (\text{J.7})$$

$$\text{Local surface variation: } C_\lambda = \frac{\lambda_3}{\sum_{i=1}^3 \lambda_i} \quad (\text{J.8})$$

To calculate the eigenvalues, we first calculate the covariance matrix, of the point positions inside of the neighbourhood and extract the eigenvalues from it, which are then sorted in descending order. Example of how these features look on a P40 sandpaper, calculated from a neighbourhood size of  $1\text{mm}$  is given in Figure J.4.

The third type of features are the statistical shape distribution features. These distributions are derived from the work by [50], for parameterization



**Fig. J.4:** Example local covariance features, extracted from the P40 sandpaper grit size. Just a small part of the sandpaper leading edge is shown for easier visualization.

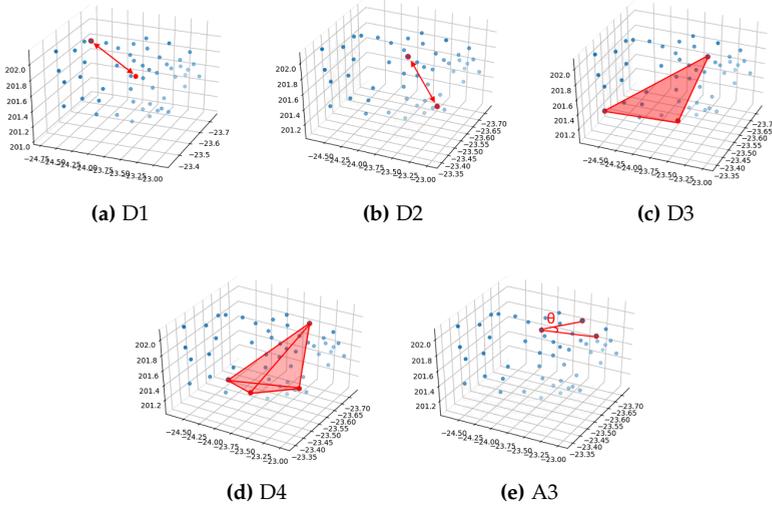
of the whole object's shape, but for use as local features, they can be used on neighbourhoods of points. These distributions are calculated as histograms of randomly sampled shape values. These shape values are based on five metrics:

- **D1** - distance from the neighbourhood centroid, to a random point from the same neighbourhood J.5a,
- **D2** - distance between two random points from the neighbourhood J.5b,
- **D3** - the square root of the area between three random points from the neighbourhood J.5c,
- **D4** - the cubic root of the volume of a tetrahedron made from four random points from the neighbourhood J.5d,
- **A3** - the angle between three random points from the neighbourhood J.5e.

We follow the suggestions by [35] and calculate the distributions as histograms with 10 bins and 255 random samplings from each neighbourhood scale. These results in 100 statistical distribution features per point. The visualization of the five used metrics, can be seen in Figure J.5.

These features are extracted from each of the reconstructed sandpaper patches and used to train a Random Forest classifier. The same classifier is used in the work of [35] and is proven to provide good results, when used with these features.

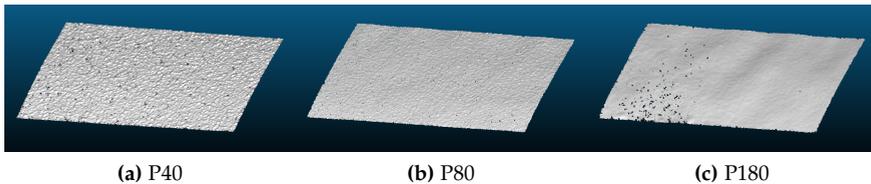
## J.4. Experimental Scenarios and Results



**Fig. J.5:** The five density metrics, used to calculate the statistical shape distributions, as proposed by [50]

## J.4 Experimental Scenarios and Results

Two testing scenarios are created as part of the paper. The first one is an initial simple classification scenario, which uses test reconstructed sandpaper patches. These patches are then classified by the trained Random Forest classifier, to prove that the selected features can describe the surface of sandpaper patches of different grit size and is selective enough to be used to describe the surfaces of wind turbine blades. The second testing scenario uses decommissioned wind turbine blades and classifies their surface depending on how close it resembles different sand paper grits.



**Fig. J.6:** The three sandpaper patches, used as an initial test for how good are the used features for describing the sandpaper surface.

**Table J.2:** Results from the initial sandpaper data test. The known grit sizes are given horizontally and the predicted grit classes are given vertically. It can be seen that highest percentage prediction for each corresponds to the real grit size of the patch.

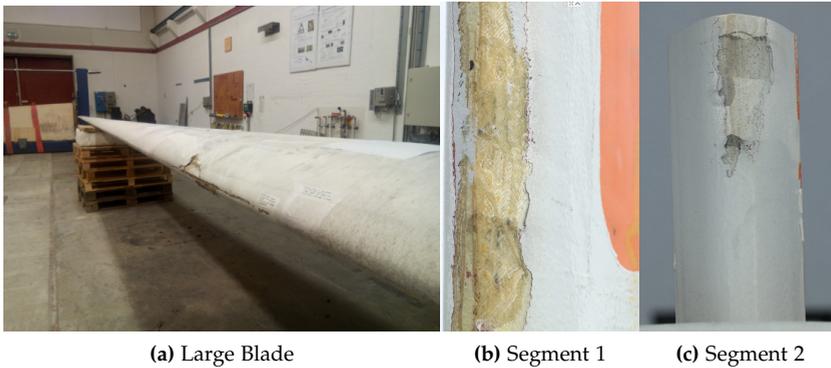
Grit Size	P40	P80	P180
P40	<b>0.761</b>	0.013	0.002
P60	0.193	0.127	0.101
P80	0.029	<b>0.696</b>	0.010
P100	0.007	0.055	0.059
P120	0.003	0.104	0.253
P180	0.007	0.005	<b>0.575</b>

To ensure that the training and testing data do not come from the same sandpaper samples, the testing sandpapers for the first scenario are selected from a different manufacturer. This testing data represents three grit sizes - a rough grit size P40, a medium grit size P80 and a fine grit size P180. The three sandpaper patches are reconstructed using Metashape, using the same capturing protocol given in Subsection J.3.2. The resultant reconstructions are shown in Figure J.6. These reconstructions are then used as testing data, to determine if the chosen features could describe unknown sandpaper surfaces, before transferring the knowledge to another type of surface. The resultant confusion matrix is given in Table J.2. It can be seen that the highest percentage prediction for each of the testing sandpaper patches, showed in bold, corresponds to the real grit size of that patch. Some of the other grit sizes are also detected, as each of the patches contains smoother and rougher areas. For the P180, some problems come from the possible present reconstruction noise and low frequency surface roughness (Figure J.6c), which can be classified as higher grit sizes.

For the second testing scenario three wind turbine blades, with varying surface roughness are selected. All the selected blades have been decommissioned and contain both smooth and very rough and damaged surface patches. One is a full blade, from which a number of patches are selected for reconstruction and the other two are smaller blade segments, which are used whole for the experiments. Four patches are selected from the full blade - two representing damaged and very rough areas and two representing relatively clean areas, with small amounts of surface roughness. The large blade, together with the two blade segments are shown in Figure J.7.

The four patches and two blade segments are reconstructed using Metashape, following the same capturing protocol presented in Subsection J.3.2. The selected features, presented in J.3.3, are extracted from the blade reconstructions and used as testing data for the Random Forest classifier. The result point clouds with pseudocolor information, on which sandpaper roughness best describes each point is given in Figure J.8. The P40 grit size is

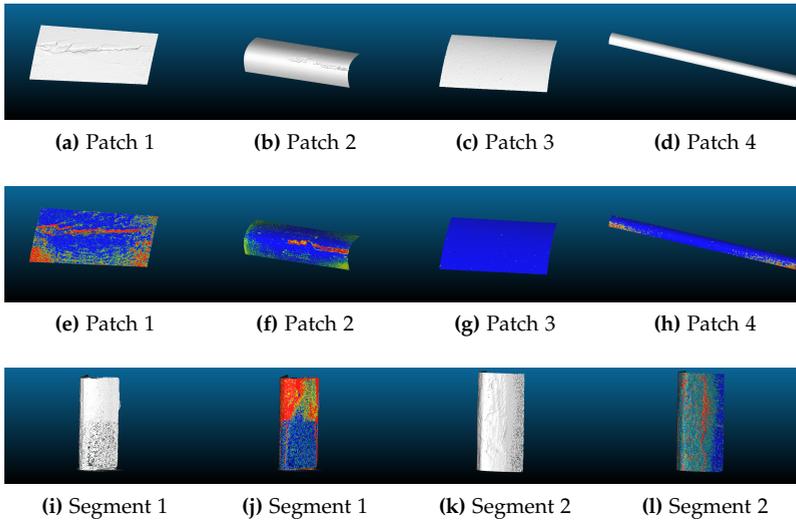
#### J.4. Experimental Scenarios and Results



**Fig. J.7:** The three wind turbine blades used for the second experiment. Four patches with varying degree of roughness are selected from the first large blade, while the other two segments are used whole

shown as red color and the P180 as dark blue color, with all other grit sizes represented with the in-between colors.

As there is no ground truth for calculating the accuracy of the surface classification, its quality will be represented in a different way. Each blade surface is separated into spherical neighbourhoods and a plane is fit to the points inside these neighbourhoods. The roughness is then calculated as the distance to this best fitting plane. The radius of the spherical neighbourhood is heuristically selected to best capture the roughness of the whole blade point cloud. The process is done through the roughness calculation functionality of CloudCompare [51]. The distance between this roughness and the calculated surface classification into sandpaper roughness is then computed for each point. For this purpose the classified sandpaper roughness for each point is substituted for the nominal average diameter of the sandpaper grains given in Table J.3.2. To quantify these distances we then calculate their root mean square error (RMSE) and standard deviation, which would give an overview of how good the sandpaper roughness is fit to the blade surface. These values are presented in Table J.3. It can be seen that the damaged surface parts are mostly classified as a P40 grit, with the parts that have varying degrees of roughness classified as the smaller grit sizes. The P60 and P120 have the least amount of point classified as them. The smooth and clean surfaces, especially on the two patches (Figure J.8g and J.8h) and on the blade segments are represented with P180 grit size. From Table J.3 it can be seen that on average the rougher patches exhibit a higher RMSE between the sandpaper representation and the real roughness. The smoother patches have less errors in the representation and a less standard deviation in these errors. The blade



**Fig. J.8:** Results from segmentation of the wind turbine blades, visualized in pseudocolor, where P40 grit is shown as red and P180 is dark blue, while the other grit sizes are the colors between them. Just the point cloud is also given for each, for easier visualization of the roughness.

segments on Figure J.8j has a larger RMSE value, because the damages on it exhibit roughness values, which far exceed the values of the sandpaper grits.

**Table J.3:** Results from comparing the sandpaper roughness value given to each point, to the roughness calculated by measuring the distance of the point to the best fitting plane in an area around it

	RMSE [ $mm$ ]	Std. Dev. [ $mm$ ]
Patch 1	0.179	0.113
Patch 2	0.125	0.069
Patch 3	0.070	0.009
Patch 4	0.101	0.060
Segment 1	0.162	0.110
Segment 2	0.288	0.122

## J.5 Conclusion

In this paper we presented an idea for segmenting wind turbine blade surfaces depending on the sandpaper grit size that best represents their roughness. This solution aims to provide a standardized method classifying surface

roughness of wind turbine blades that can be used for calculating their energy output and performance, as well as more easily modeling them for tests in wind tunnels.

We selected six different sandpaper patches with varying grit sizes and 3D reconstructed them using SfM. We then extracted a number of geometrical, covariance and statistical features from neighbourhoods with progressively smaller sizes. We used these features to train a Random Forest classifier.

To test the proposed solution we first evaluated the classifier on a testing set of sandpaper patches. We verified that the extracted features could be used to identify each grit size of the training set. We then introduced surface data from three wind turbine blades. The data represented surfaces with varying degrees of surface roughness and damages. These surfaces were also 3D reconstructed and then used as a testing dataset. We demonstrated that we Random Forest classifier managed to sufficiently segment the surfaces and to represent their roughness as sandpaper grit sizes.

To give a definitive answer if quantifying the surface roughness of wind turbine blades is possible using sandpapers of different grit sizes, a number of wind tunnel experiments need to be made. A blade with certain surface roughness would be tested against an equivalent blade, with sandpapers of grit sizes representing the same roughness profile. The aerodynamic properties of each of them would need to be computed and compared.

## References

- [1] H. S. El-Mesery, H. Mao, and A. E.-F. Abomohra, "Applications of non-destructive technologies for agricultural and food products quality inspection," *Sensors*, vol. 19, no. 4, p. 846, 2019.
- [2] C. Banica, S. V. Paturca, S. D. Grigorescu, and A. M. Stefan, "Data acquisition and image processing system for surface inspection," in *2017 10th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*. IEEE, 2017, pp. 28–33.
- [3] R. Dastoorian, A. E. Elhabashy, W. Tian, L. J. Wells, and J. A. Camelio, "Automated surface inspection using 3d point cloud data in manufacturing: A case study," in *International Manufacturing Science and Engineering Conference*, vol. 51371. American Society of Mechanical Engineers, 2018, p. V003T02A036.
- [4] Y. Zhang, X. Yuan, Y. Fang, and S. Chen, "Uav low altitude photogrammetry for power line inspection," *ISPRS International Journal of Geo-Information*, vol. 6, no. 1, p. 14, 2017.

## References

- [5] R. W. Martin, A. Sabato, A. Schoenberg, R. H. Giles, and C. Niezrecki, "Comparison of nondestructive testing techniques for the inspection of wind turbine blades' spar caps," *Wind Energy*, vol. 21, no. 11, pp. 980–996, 2018.
- [6] Y. Du, S. Zhou, X. Jing, Y. Peng, H. Wu, and N. Kwok, "Damage detection techniques for wind turbine blades: A review," *Mechanical Systems and Signal Processing*, vol. 141, p. 106445, 2020.
- [7] X.-l. Li, J. Sun, N. Tao, L. Feng, J.-l. Shen, Y. He, C. Zhang, and Y.-j. Zhao, "An effective method to inspect adhesive quality of wind turbine blades using transmission thermography," *Journal of Nondestructive Evaluation*, vol. 37, no. 2, p. 19, 2018.
- [8] C. M. Langel, R. Chow, O. F. Hurley, C. C. P. Van Dam, D. C. Maniaci, R. S. Ehrmann, and E. B. White, "Analysis of the Impact of Leading Edge Surface Degradation on Wind Turbine Performance," in *33rd Wind Energy Symposium*. Reston, Virginia: American Institute of Aeronautics and Astronautics, jan 2015. [Online]. Available: <http://arc.aiaa.org/doi/10.2514/6.2015-0489>
- [9] M. Schramm, H. Rahimi, B. Stoevesandt, and K. Tangager, "The Influence of Eroded Blades on Wind Turbine Performance Using Numerical Simulations," *Energies*, vol. 10, no. 9, p. 1420, sep 2017. [Online]. Available: <http://www.mdpi.com/1996-1073/10/9/1420>
- [10] C. Bak, M. Gaunaa, A. S. Olsen, and E. K. Kruse, "What is the critical height of leading edge roughness for aerodynamics," in *Journal of Physics: Conference Series*, vol. 753, no. 2, 2016, p. 022023.
- [11] A. Sareen, C. A. Sapre, and M. S. Selig, "Effects of leading edge erosion on wind turbine blade performance," *Wind Energy*, vol. 17, no. 10, pp. 1531–1542, oct 2014. [Online]. Available: <http://doi.wiley.com/10.1002/we.1649>
- [12] X. Chen, "Fracture of wind turbine blades in operation—part i: A comprehensive forensic investigation," *Wind Energy*, vol. 21, no. 11, pp. 1046–1063, 2018.
- [13] A. Shihavuddin, X. Chen, V. Fedorov, A. Nymark Christensen, N. Andre Brogaard Riis, K. Branner, A. BJORHOLM DAHL, and R. Reinhold Paulsen, "Wind Turbine Surface Damage Detection by Deep Learning Aided Drone Inspection Analysis," *Energies*, vol. 12, no. 4, p. 676, feb 2019. [Online]. Available: <http://www.mdpi.com/1996-1073/12/4/676>

## References

- [14] D. Denhof, B. Staar, M. Lütjen, and M. Freitag, "Automatic optical surface inspection of wind turbine rotor blades using convolutional neural networks," *Procedia CIRP*, vol. 81, pp. 1166–1170, 2019.
- [15] S. Moreno, M. Peña, A. Toledo, R. Treviño, and H. Ponce, "A new vision-based method using deep learning for damage inspection in wind turbine blades," in *2018 15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*. IEEE, 2018, pp. 1–5.
- [16] A. Al-Kaff, F. M. Moreno, L. J. San José, F. García, D. Martín, A. de la Escalera, A. Nieva, and J. L. M. Garcéa, "Vbii-uav: Vision-based infrastructure inspection-uav," in *World Conference on Information Systems and Technologies*. Springer, 2017, pp. 221–231.
- [17] D. Zhang, K. Burnham, L. McDonald, C. Macleod, G. Dobie, R. Summan, and G. Pierce, "Remote inspection of wind turbine blades using uav with photogrammetry payload," in *56th Annual British Conference of Non-Destructive Testing-NDT 2017*, 2017.
- [18] D. Zhang, R. Watson, G. Dobie, C. MacLeod, A. Khan, and G. Pierce, "Quantifying impacts on remote photogrammetric inspection using unmanned aerial vehicles," *Engineering Structures*, p. 109940, 2020.
- [19] M. Nielsen, I. Nikolov, E. Kruse, J. Garnæs, and C. Madsen, "High-resolution structure-from-motion for quantitative measurement of leading-edge roughness," *Energies*, vol. 13, no. 15, 2020.
- [20] ISO4287, "Iso4287(1997) geometrical product specifications (gps) — surface texture: Profile method — terms, definitions and surface texture parameters," <https://www.iso.org/standard/10132.html>, 1997, accessed: 2020-08-03.
- [21] ISO25178, "Iso25178(2012) geometrical product specifications (gps) — surface texture: Areal — part 2: Terms, definitions and surface texture parameters," <https://www.iso.org/obp/ui/#iso:std:iso:25178:-2:ed-1:v1:en>, 2012, accessed: 2020-08-03.
- [22] A. P. Broeren and M. B. Bragg, "Effect of airfoil geometry on performance with simulated intercycle ice accretions," *Journal of Aircraft*, vol. 42, no. 1, pp. 121–130, 2005.
- [23] O. F. Marzuki, A. S. M. Rafie, F. I. Romli, and K. A. Ahmad, "Magnus wind turbine: the effect of sandpaper surface roughness on cylinder blades," *Acta Mechanica*, vol. 229, no. 1, pp. 71–85, 2018.
- [24] M. S. Genc, K. Kemal, and H. H. Açikel, "Investigation of pre-stall flow control on wind turbine blade airfoil using roughness element," *Energy*, vol. 176, pp. 320–334, 2019.

## References

- [25] "Iso 6344(1998) coated abrasives — grain size analysis parts 1-3," <https://www.iso.org/standard/12643.html>, accessed: 2019-03-06.
- [26] W. Han, J. Kim, and B. Kim, "Effects of contamination and erosion at the leading edge of blade tip airfoils on the annual energy production of wind turbines," *Renewable energy*, vol. 115, pp. 817–823, 2018.
- [27] L. Wang and Z. Zhang, "Automatic detection of wind turbine blade surface cracks based on uav-taken images," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 9, pp. 7293–7303, 2017.
- [28] N. Gaudern, "A practical study of the aerodynamic impact of wind turbine blade leading edge erosion," in *Journal of Physics: Conference Series*, vol. 524, no. 1. IOP Publishing, 2014, p. 012031.
- [29] C. M. Langel, R. Chow, C. Van Dam, and D. C. Maniaci, "Rans based methodology for predicting the influence of leading edge erosion on airfoil performance," Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2017.
- [30] D. Xu, C. Wen, and J. Liu, "Wind turbine blade surface inspection based on deep learning and uav-taken images," *Journal of Renewable and Sustainable Energy*, vol. 11, no. 5, p. 053305, 2019.
- [31] L. Peng and J. Liu, "Detection and analysis of large-scale wt blade surface cracks based on uav-taken images," *IET Image Processing*, vol. 12, no. 11, pp. 2059–2064, 2018.
- [32] C. H. L. Chan, Q. Wang, R. Holden, S. Huang, and W. Zhao, "Optimal number of control points for fitting b-splines in wind turbine blade measurement," *International Journal of Precision Engineering and Manufacturing*, vol. 20, no. 9, pp. 1507–1517, 2019.
- [33] I. Amenabar, A. Mendikute, A. López-Arraiza, M. Lizaranzu, and J. Aurrekoetxea, "Comparison and analysis of non-destructive testing techniques suitable for delamination inspection in wind turbine blades," *Composites Part B: Engineering*, vol. 42, no. 5, pp. 1298–1305, 2011.
- [34] N. Brodu and D. Lague, "3d terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 68, pp. 121–134, 2012.
- [35] M. Weinmann, B. Jutzi, S. Hinz, and C. Mallet, "Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 105, pp. 286–304, 2015.

## References

- [36] T. Hackel, J. D. Wegner, and K. Schindler, "Fast semantic segmentation of 3d point clouds with strongly varying density," *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, vol. 3, pp. 177–184, 2016.
- [37] A. Dittrich, M. Weinmann, and S. Hinz, "Analytical and numerical investigations on the accuracy and robustness of geometric features extracted from 3d point cloud data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 126, pp. 195–208, 2017.
- [38] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer, "A survey of structure from motion\*." *Acta Numerica*, vol. 26, pp. 305–364, 2017.
- [39] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [40] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.
- [41] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *International workshop on vision algorithms*. Springer, 1999, pp. 298–372.
- [42] FEPA, "Fepa - federation of european producers of abrasives," <https://www.fepa-abrasives.com/abrasive-products/grains>, 1955, accessed: 2019-03-06.
- [43] I. Nikolov and C. Madsen, "Benchmarking close-range structure from motion 3d reconstruction software under varying capturing conditions," in *Euro-Mediterranean Conference*. Springer, 2016, pp. 15–26.
- [44] Agisoft, "Metashape (photoscan)," <http://www.agisoft.com/>, 2010, accessed: 2020-06-15.
- [45] I. Nikolov and C. Madsen, "Rough or noisy? metrics for noise detection in sfm reconstructions," accepted at MDPI Sensors Journal.
- [46] R. Blomley, B. Jutzi, and M. Weinmann, "Classification of airborne laser scanning data using geometric multi-scale features and different neighbourhood types." *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 3, no. 3, 2016.
- [47] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.
- [48] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.

## References

- [49] K. F. West, B. N. Webb, J. R. Lersch, S. Pothier, J. M. Triscari, and A. E. Iverson, "Context-driven automated target detection in 3d data," in *Automatic Target Recognition XIV*, vol. 5426. International Society for Optics and Photonics, 2004, pp. 133–143.
- [50] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Transactions on Graphics (TOG)*, vol. 21, no. 4, pp. 807–832, 2002.
- [51] D. Girardeau-Montaut, "Cloudcompare-open source project," *Open-Source Project*, 2011.



## SUMMARY

This thesis focuses on the problems surrounding Structure from Motion (SfM) data capturing and analysis in the context of 3D surface inspection. The use of 3D point clouds and meshes has become more widely spread, because of the additional level of information that they contain compared to more traditionally used inspection data like 2D images. Through the thesis, each part of the SfM pipeline is examined. From looking at best practices when capturing images both terrestrial and in the air using unmanned aerial vehicles and developing simulation environments for testing capturing setups. Through the development of automatic methods for detecting the absolute scale of the reconstructed surfaces, using additional information from GPS and LiDAR data. Finally, analyzing the resultant reconstructions to determine their accuracy, the presence of noise, and possible use cases, like conveying the surface profile through vibrations and segmenting it into different roughness areas.

ISSN (online): 2446-1628

ISBN (online): 978-87-7210-818-6 **AALBORG UNIVERSITY PRESS**